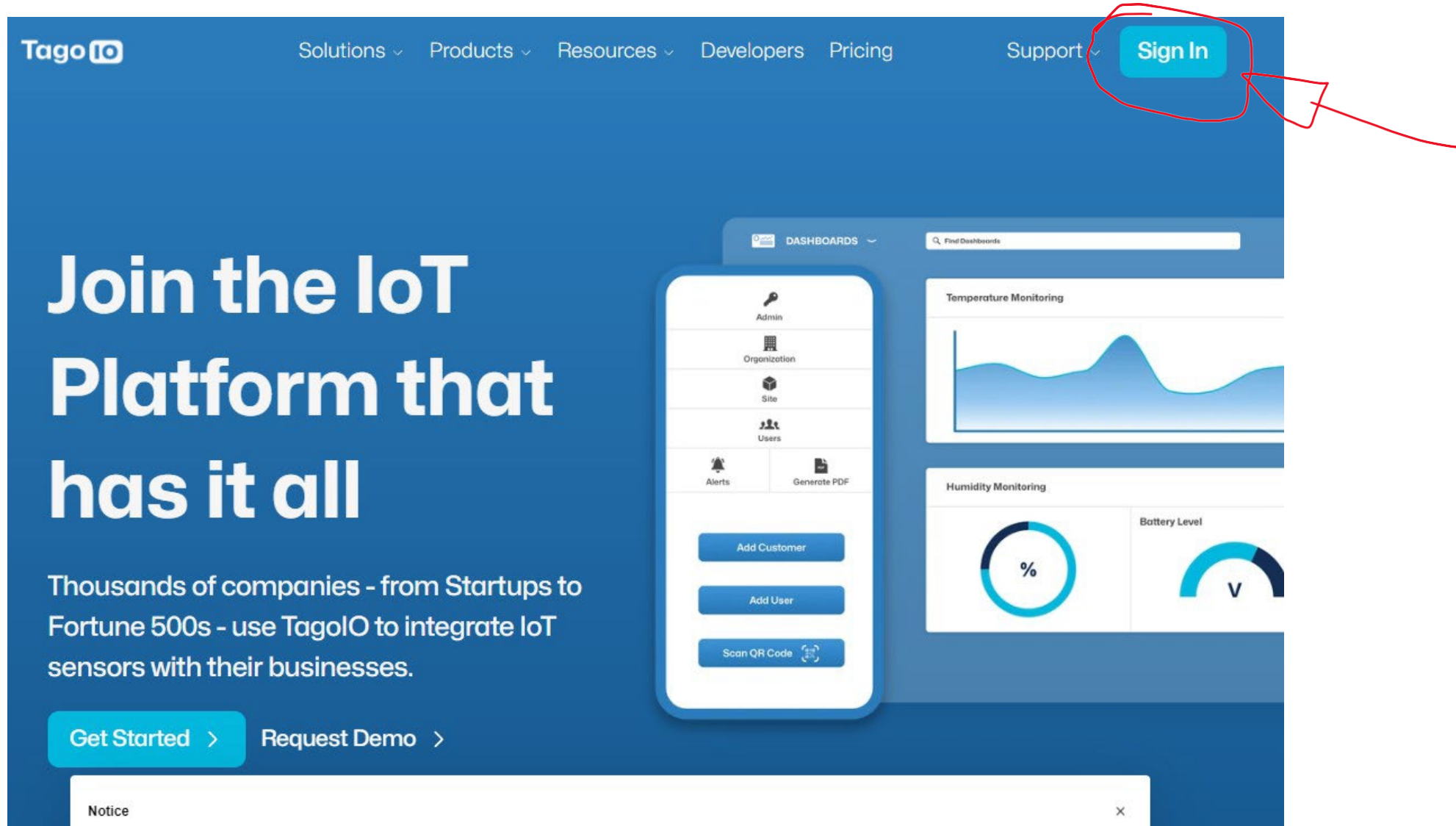


Criando uma conta na Tago Io



The screenshot displays the Tago.io website interface. The top navigation bar includes the Tago.io logo, links for Solutions, Products, Resources, Developers, Pricing, and Support, and a prominent blue 'Sign In' button which is circled in red with an arrow pointing to it. The main content area features a large blue background with the text 'Join the IoT Platform that has it all'. Below this, a message states: 'Thousands of companies - from Startups to Fortune 500s - use TagoIO to integrate IoT sensors with their businesses.' At the bottom of this section are two buttons: 'Get Started >' and 'Request Demo >'. On the right side, there is a preview of the Tago.io dashboard, showing a sidebar with options like Admin, Organization, Site, Users, Alerts, and Generate PDF, and a main area with charts for Temperature Monitoring, Humidity Monitoring, and Battery Level. A 'Notice' banner is visible at the very bottom of the page.

Tago.io

Solutions ▾ Products ▾ Resources ▾ Developers Pricing Support ▾ **Sign In**

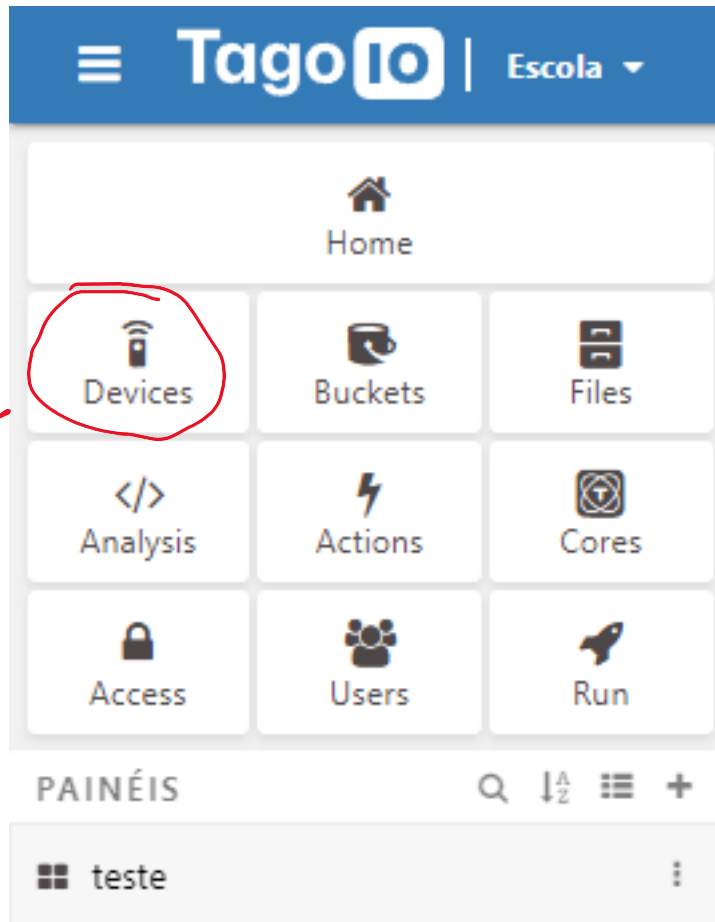
Join the IoT Platform that has it all

Thousands of companies - from Startups to Fortune 500s - use TagoIO to integrate IoT sensors with their businesses.

Get Started > **Request Demo >**

Notice

Adicionando um Device



Custom MQTT

- Simulator
- AWS IoT
- BeWhere
- Cellio
- Generic Endpoint
- HTTP
- HTTPS
- Kinéis
- LOKA
- LoRaWAN Actility
- LoRaWAN ChirpStack
- LoRaWAN CityKinect
- LoRaWAN Everynet
- LoRaWAN Helium
- LoRaWAN Kerlink
- LoRaWAN Loriot
- LoRaWAN MachineQ
- LoRaWAN Orbiwise

Learn more →

search a connector for your device

All networks ▼

Last used

- Custom MQTT
Connect any device using the MQTT protocol t...

Recently added

- Custom HTTPS
Connect any device using HTTPS protocol directly to send/get data
- Custom MQTT
Connect any device using the MQTT protocol to send/get data
- Arduino
Connect your Arduino board directly to TagIO
- Custom Sigfox
Use this Custom setup if your device connected through SIGFOX doesn't...
- Custom Everynet
Use this custom connector if your device using Everynet doesn't show up in the...



MQTT: The Standard for IoT Messaging

Custom MQTT

Details

Give a name for this device and learn about this network [here](#).

Define the type of bucket to be used to store data for this device.

Device name

Device #1

Data storage type

Device Data Optimized (Immutable)

Data Retention

The Data Retention feature automatically removes old data from the bucket after the period you define here. [Learn more](#).

Period

Monthly

Retention

1

This selection limits the storage for this device to 1 Million data registers per month. [Upgrade your plan](#) to change the Period and Retention.

Main information

Set the initial configurations for this device.

Payload Type

Auto Parser (JSON or TEXT)

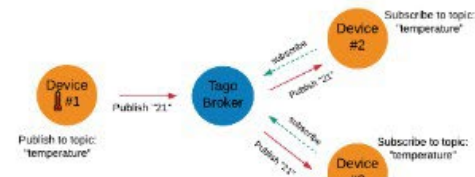
Description

Learn how this device works, and what its capabilities are.

This is quick setup to create new devices that can be used for any applications and functions to access the TagoIO MQTT broker.

Your device will only need to use the device-token to send data to your account. Don't use the Authorization feature when working directly our broker.

More information can be found in the [documentation link](#) shown above.



Cancel

Create my Device

FIAP



MQTT: The Standard for IoT Messaging

Custom MQTT



All done!

- ✓ Creating device
- ✓ Creating and linking a bucket

Create another device

Finish

FIAP



AulaTeste

Last Input Never | Type Device Data Optimized (Immutable)

General Information

Emulator

Payload Parser

● Live Inspector

Configuration Parameters

Tags

More

⚙️ General Information

📱 Name

AulaTeste



☰ Token & Serial Number

Token Name

Token #2

Generate

Default



🌐 Network

MQTT

🔌 Connector

Custom MQTT

Token information



Token

Name

Default

Expires at

Never

Fechar

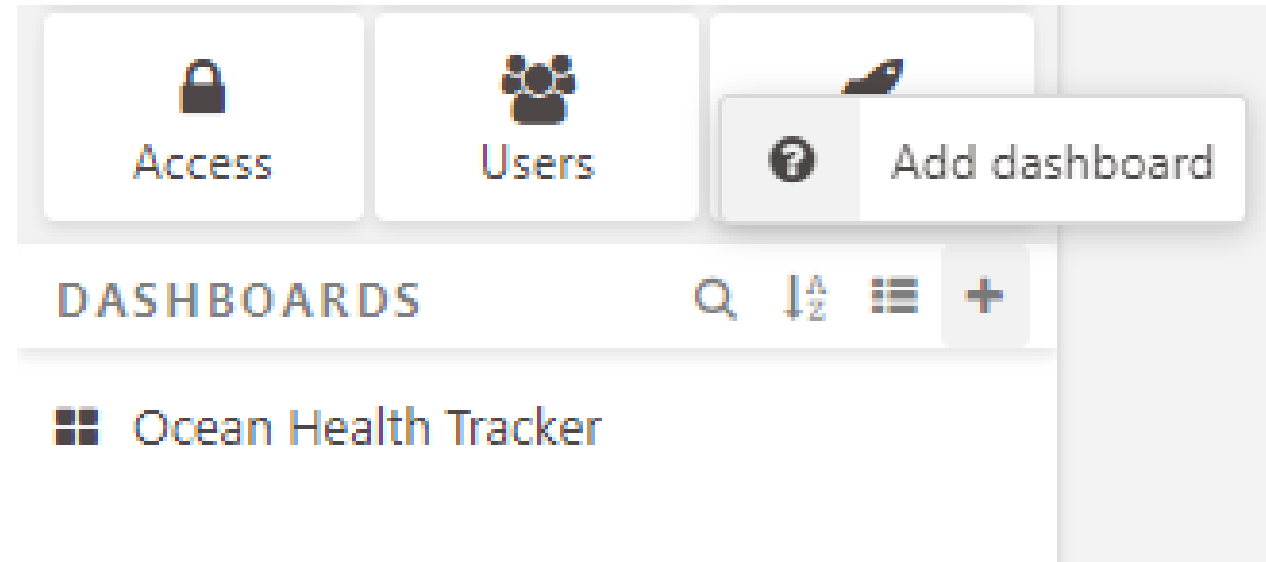
Copy Token

COLE AQUI SEU TOKEN E INICIE O WOKWI

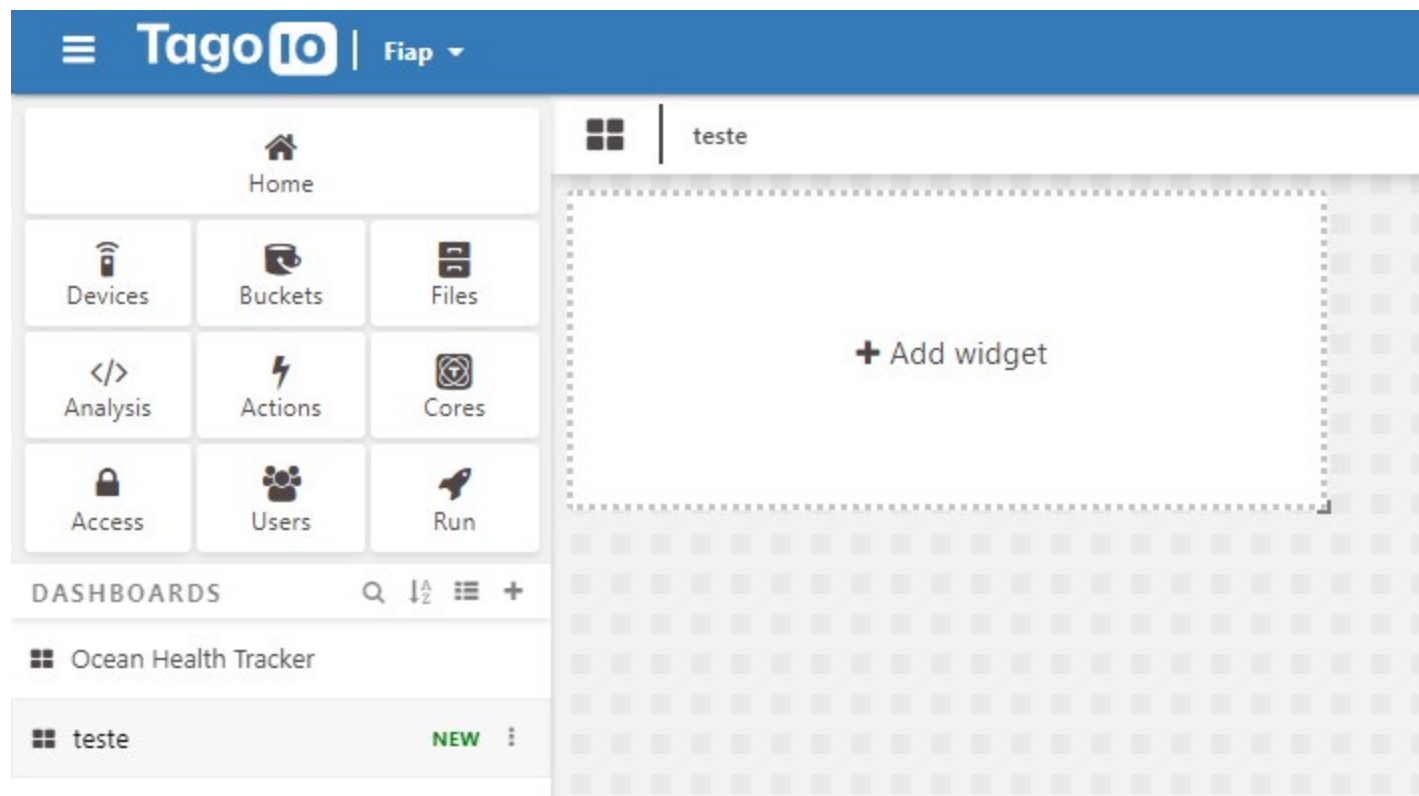
```
// Configurações do TagoIO
const char* token_dispositivo = [REDACTED] // Insira seu token do dispositivo TagoIO aqui
const char* servidor = "https://api.tago.io/data";

// Configurações dos pinos
#define PINO_DHT 17 // DHT22 (ajustado para evitar conflito)
#define TIPO_DHT DHT22
const int pino_trig = 23; // HC-SR04 Trig
const int pino_echo = 19; // HC-SR04 Echo
```


CRIE UMA DASHBOARD COM O NOME “OCEAN HEALTH TRACKER”



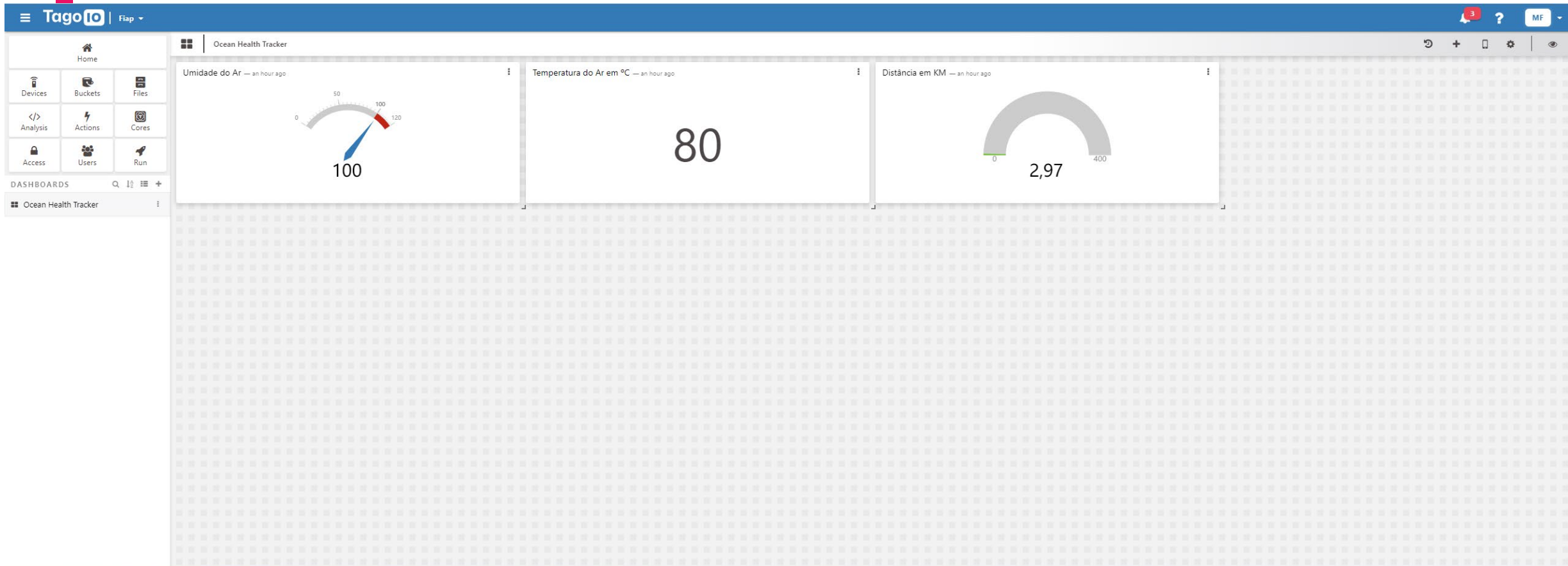
PARA ADICIONAR OS VALORES QUE ESTÃO SENDO MOSTRADOS NO WOKWI, BASTA ADICIONAR UM WIDGET CLICANDO EM “ADD WIDGET”



PARA TRAZER OS VALORES DO ESP32, BASTA ESCOLHER O WIDGET QUE VOCÊ SENTE MAIS CONFORTÁVEL DE LER E PRESSIONAR EM “**DATA FROM**” SELECIONAR O NOME DO DEVICE QUE VOCÊ CRIOU ANTERIORMENTE E A VARIÁVEL QUE VOCÊ QUER USAR OU CHAMAR! NÃO ESQUEÇA DE APERTAR O BOTÃO “**SAVE**” E PRONTO! SÓ FAZER O MESMO PARA A OUTRA VÁRIAVEL

The image shows the configuration interface for a 'VuMeter' widget. The 'Data from' section is highlighted with a red box, indicating the selection of the data source. Below this, the 'Device' is set to 'OceanHealthTracker' and the 'Variable' is set to 'umidadeair'. The 'Save' button at the bottom right is highlighted with a green box, indicating the final step in the configuration process.

O RESULTADO FINAL É ESSE!



Copyright © 2024 Prof. Yan Coelho e Equipe Ocean Health Tracker

Todos direitos reservados. Reprodução ou divulgação total ou parcial deste documento é expressamente proibido sem o consentimento formal, por escrito, do Professor (autor).