

# Automação para longa distância

**ESP32 SIM800L Firebase App Fernando K e Relê**



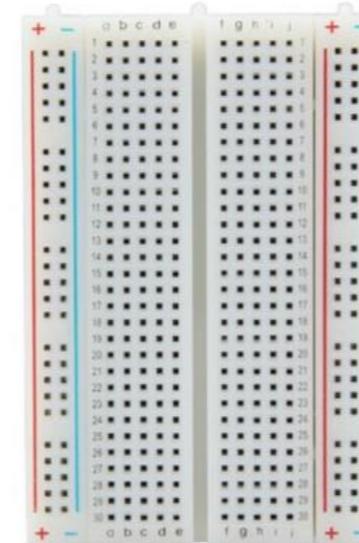
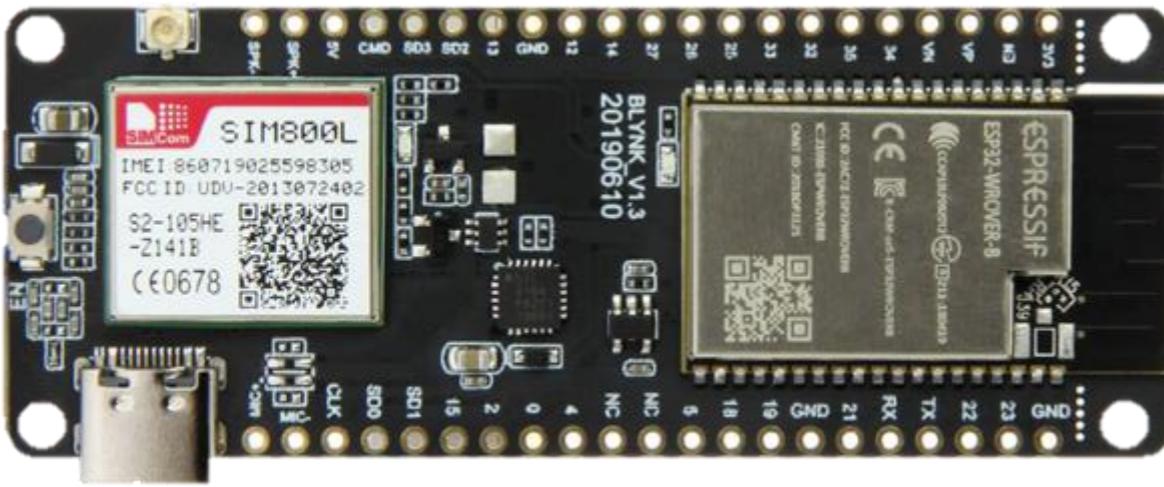
Por Fernando Koyanagi

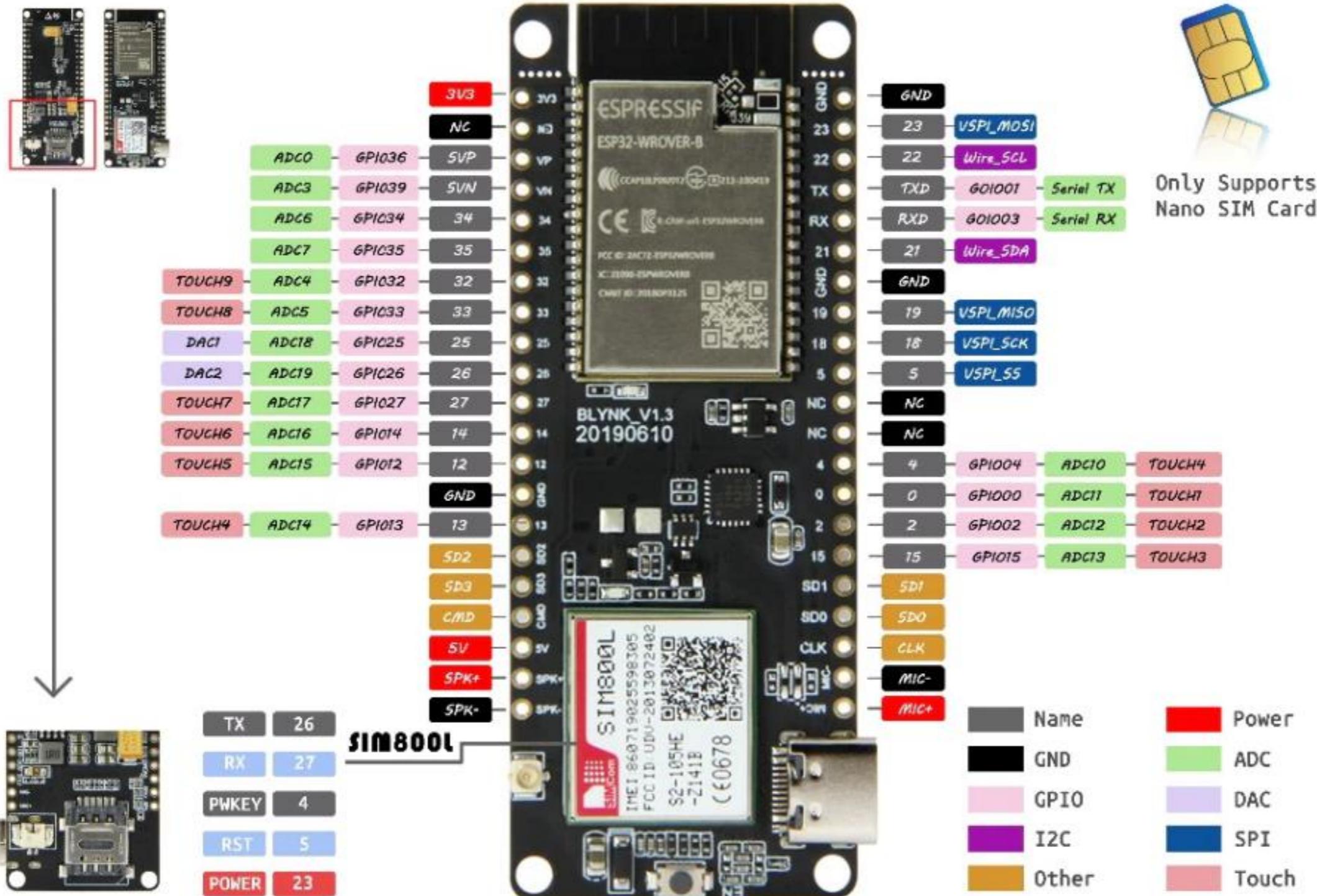
# **Intenção dessa aula**

**Utilizando o app Fernando K e o Firebase, ligar  
e desligar um relê que está em um ESP32  
conectado à rede de dados de um SIM Card**

# Recursos usados

- TTGO LILYGO T-Call ESP32 v1.3
- Protoboard
- Jumpers
- Relé





ESP32 SIM Card SIM800L T-Call V1.3



## Robô de desenho XY

by Fernando K - 19 fevereiro

Temos aqui hoje um projeto de mecatrônica que é uma derivação de um vídeo que eu já lancei aqui: ROUTER E PLOTTER WIFI COM WEB SERVER ...

[Leia mais](#)



## E se o seu link cair?

by Fernando K - 12 fevereiro

Neste vídeo vamos criar um sensor de queda de link com um ESP32 e um SIM800. Isso significa que, com este projeto, poderemos verificar ...

[Leia mais](#)



## Tragédia de Brumadinho: sugestão para um sistema de alerta!

by Fernando K - 08 fevereiro

Como a Internet das Coisas pode auxiliar em tragédias como a de Mariana e Brumadinho? Hoje quero trazer uma sugestão de um sistema ...

[Leia mais](#)



## Router e Plotter WiFi com Webserver em ESP32

by Fernando K - 05 fevereiro

Já gravei vídeos sobre plotter com o Raspberry Pi e com Laser, mas, hoje, quero falar de uma versão com GRBL e ESP32. Como acredito e...

[Leia mais](#)



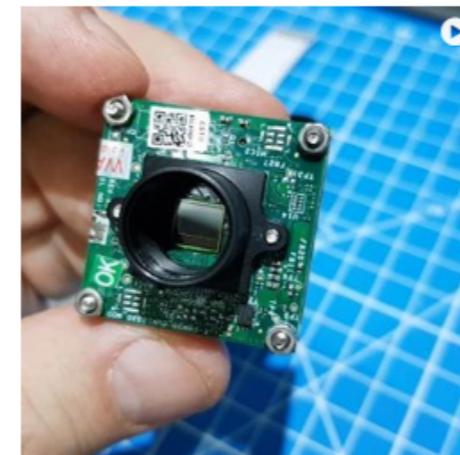
# Links onde comproi os componentes

Em [www.fernandok.com](http://www.fernandok.com)

Supergrupo de colaboração entre meus seguidores.

Conteúdo exclusivo, que não tem no Youtube!

### INSTAGRAM @FERNANDOK\_OFICIAL



### FACEBOOK



### INSCREVA-SE NO YOUTUBE

# forum.fernandok.com

Fórum Fernando K Tecnologia  
Fórum sobre dúvidas com relação ao conteúdo disponibilizado pelo Fernando Koyanagi

Nosquisar...

[www.fernandok.com](http://www.fernandok.com) /fernandokoyanagi /fernandokoyanagi /fernandok\_oficial /fernandok\_oficial

[Links rápidos](#) [L...](#) [fernandokoyanagi](#)

Bem-vindo: 05/Oct/2018, 11:16 A sua última visita foi em 10/Set/2018, 15:47

Assinalar todos os fóruns como lidos

**SUporte Fórum Fernandok**

	TÓPICOS	MENSAGENS	ÚLTIMA MENSAGEM
Feedback Dúvidas, críticas ou sugestões sobre o Fórum FernandoK. Para demais questões utilize o fórum correto.	6	11	<b>Re: O russo voltou</b> por Ipmehi 01/Oct/2018, 08:25

**FERNANDO K**

	TÓPICOS	MENSAGENS	ÚLTIMA MENSAGEM
Arduino Projetos de arduino	31	79	<b>skardy bogii</b> por Sorororcem 05/Oct/2018, 10:55
ESP32 Projetos de ESP32	29	62	<b>Dúvidas sobre como instalar a...</b> por Marcos Sarge 04/Oct/2018, 15:52
ESP8266 O ESP8266 é um microcontrolador do fabricante chinês Espressif que inclui capacidade de comunicação por Wi-Fi.	24	51	<b>Re: NodeMCU não conecta em qu...</b> por ivanribeira 04/Oct/2018, 14:39
LoRa Projetos com LoRa	11	31	<b>Projeto de irrigação de jardim</b> por marlendo 04/Oct/2018, 21:30
STM32 Projetos com STM32	3	8	<b>Re: Imprecisão de tempo de de...</b> por biazoto 12/Sep/2018, 09:15
Motor Projetos com motor	5	11	<b>Re: impressora 3d com motor dc</b> por Magneton 24/Sep/2018, 19:05
Display Projetos com Display	4	11	<b>Re: Alguém conhece o VIRTUINO...</b> por Jod Luz 21/Sep/2018, 11:39

**QUEM ESTÁ ONLINE**  
No total, há 4 usuários online :: 2 usuários registrados, 0 invitado e 2 visitantes (baseado em usuários ativos nos últimos 5 minutos)  
O recorde de usuários online foi de 19 em 11/Sep/2018, 05:37

Usuários registrados: alberto, fernandokoyanagi  
Legenda: Administradores, Moderadores globais

**ANIVERSÁRIOS**  
Não há aniversários hoje

**ESTATÍSTICAS**  
Total de mensagens 703 • Total de tópicos 114 • Total de membros 469 • Novo usuário: Sorororcem

[L...](#) [Anunciar](#)

Powered by phpBB® Forum Software © phpBB Limited  
Traduzido por: Suporte phpBB  
Painel de Controle da Administração



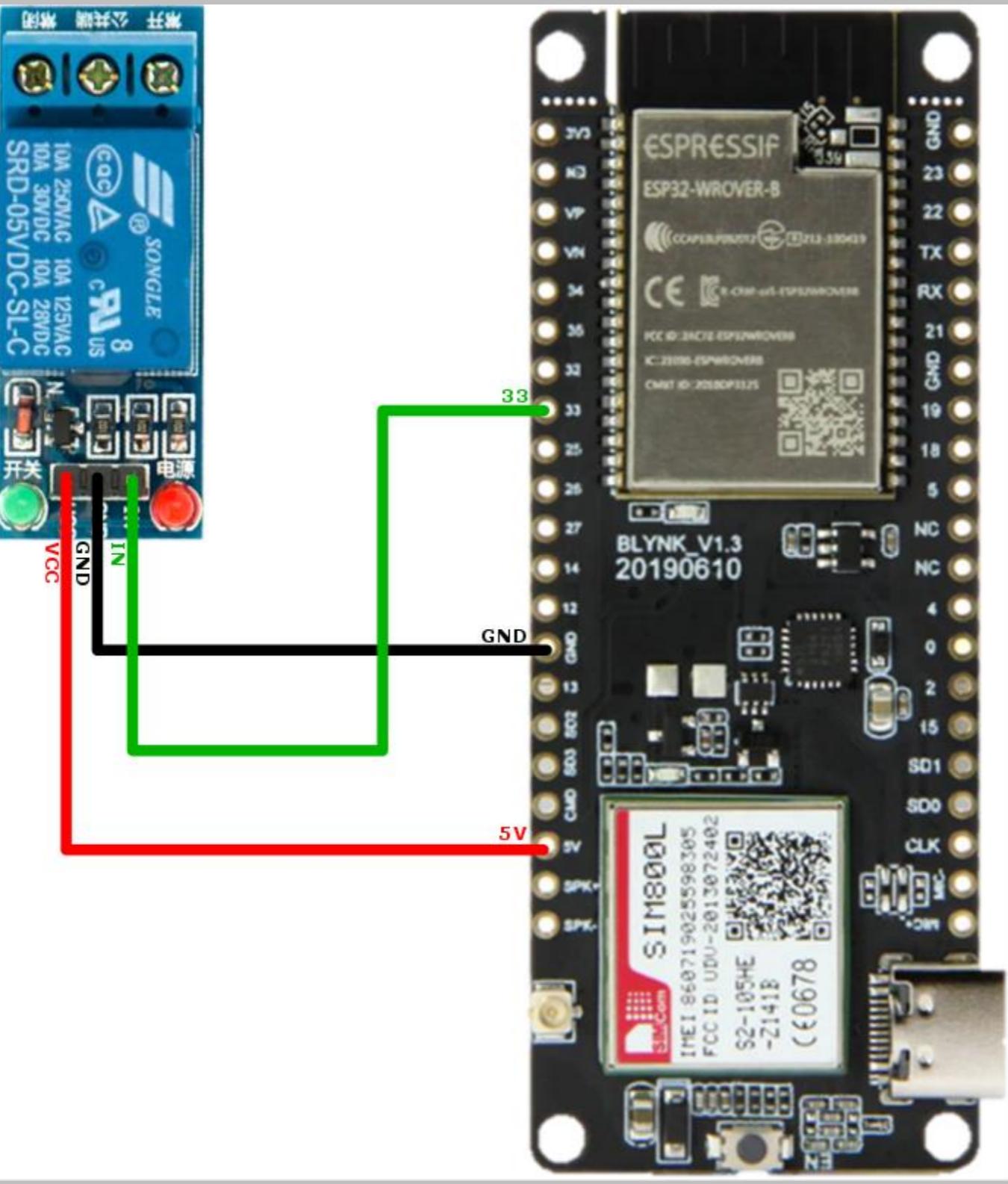
**Instagram**  
fernandok\_oficial



**Telegram**  
fernandok\_oficial



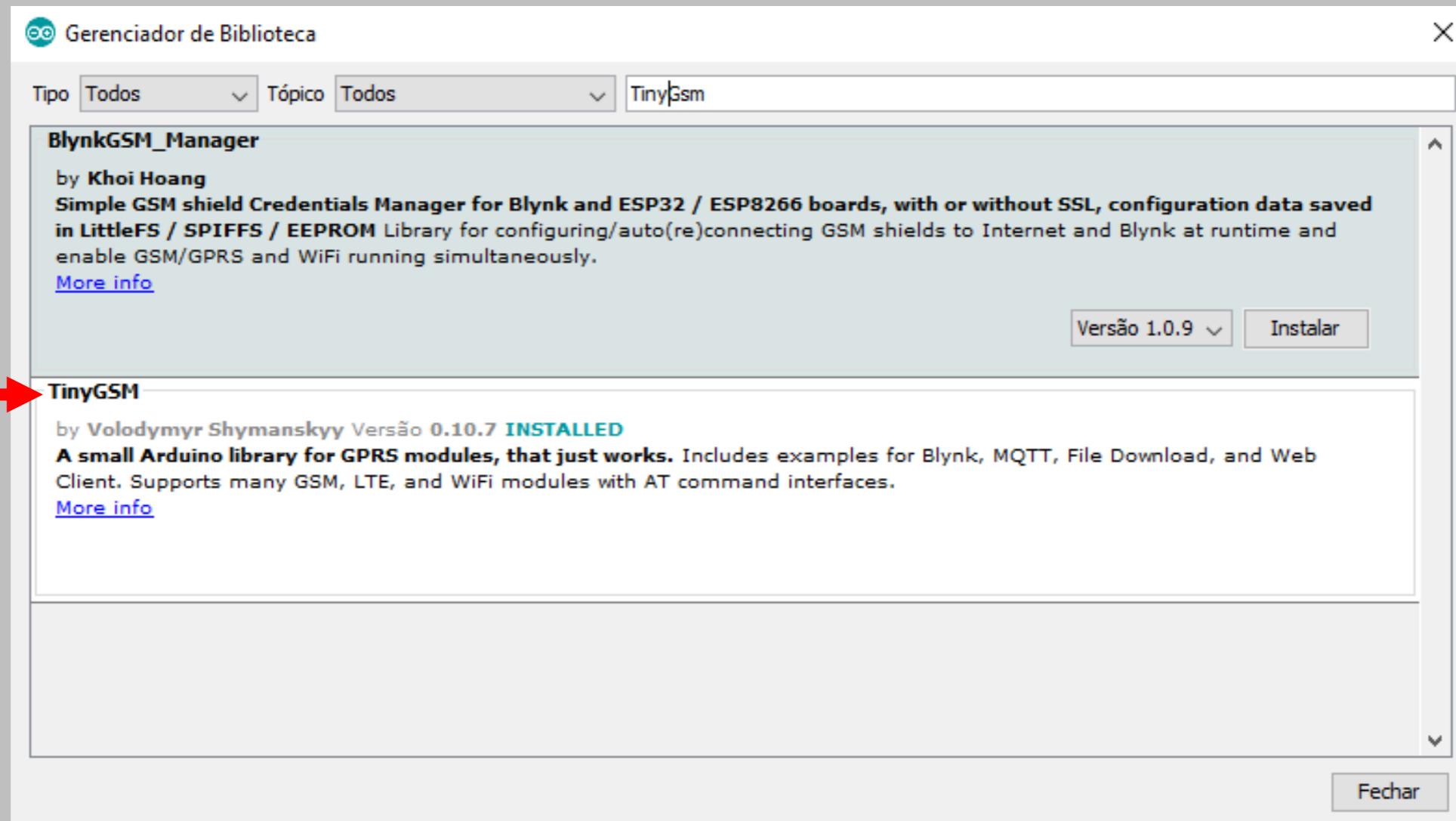
# **Montagem**



# **Bibliotecas**

# Biblioteca

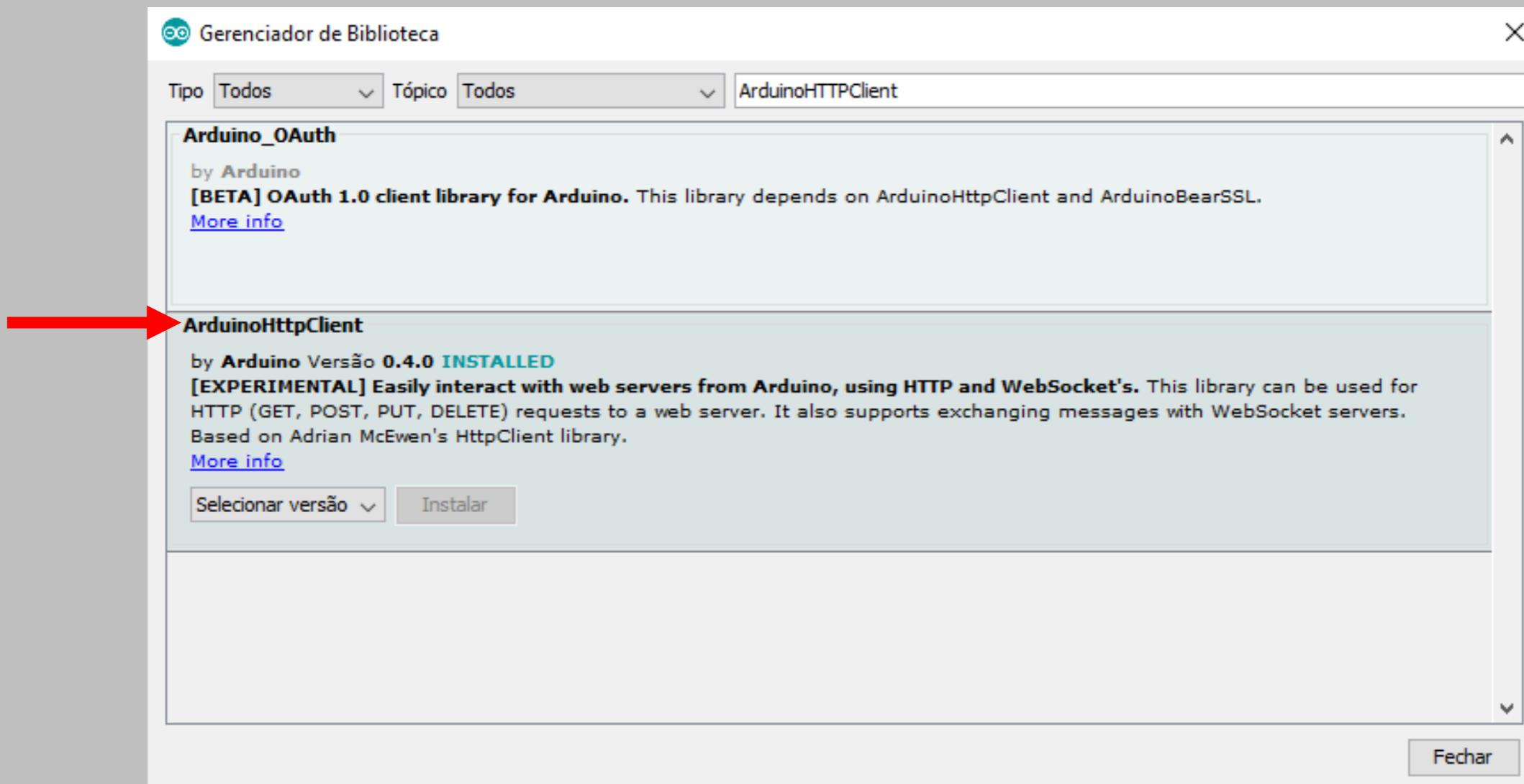
## TinyGSM



<https://github.com/vshymanskyy/TinyGSM>

# Biblioteca

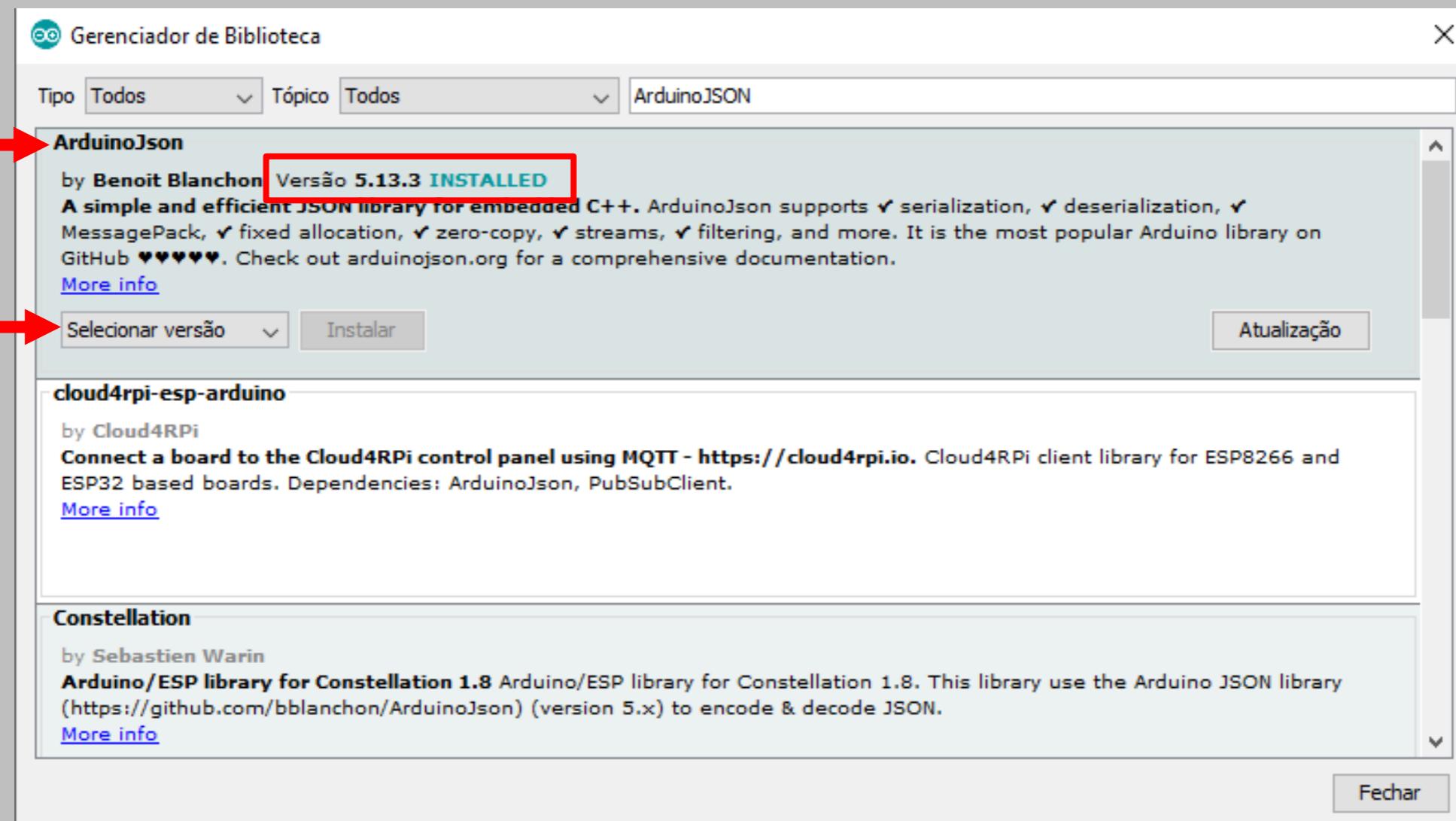
## ArduinoHTTPClient



<https://github.com/arduino-libraries/ArduinoHttpClient>

# Biblioteca

## ArduinoJson v5.13.3



Selecione a  
versão  
5.13.3 aqui  
antes de  
instalar

<https://github.com/bblanchon/ArduinoJson>

# Escolhendo a placa

ESP32 Sketch Data Upload

```
placa: "ESP32 Wrover Module"  
Upload Speed: "921600"  
lash Frequency: "80MHz"  
lash Mode: "QIO"  
artition Scheme: "Default 4MB with spiffs (1.2MB APP/1.5MB SPIFFS)"  
ore Debug Level: "Nenhum"  
orta  
Obter informações da Placa
```

rogramador: "AVRISP mkII"

gravar Bootloader

GSM) ;

```
Liente http  
Client (modemGSM) ;
```

```
inicação com o Firebase  
mClient, FIREBASE_HOST, 443);
```

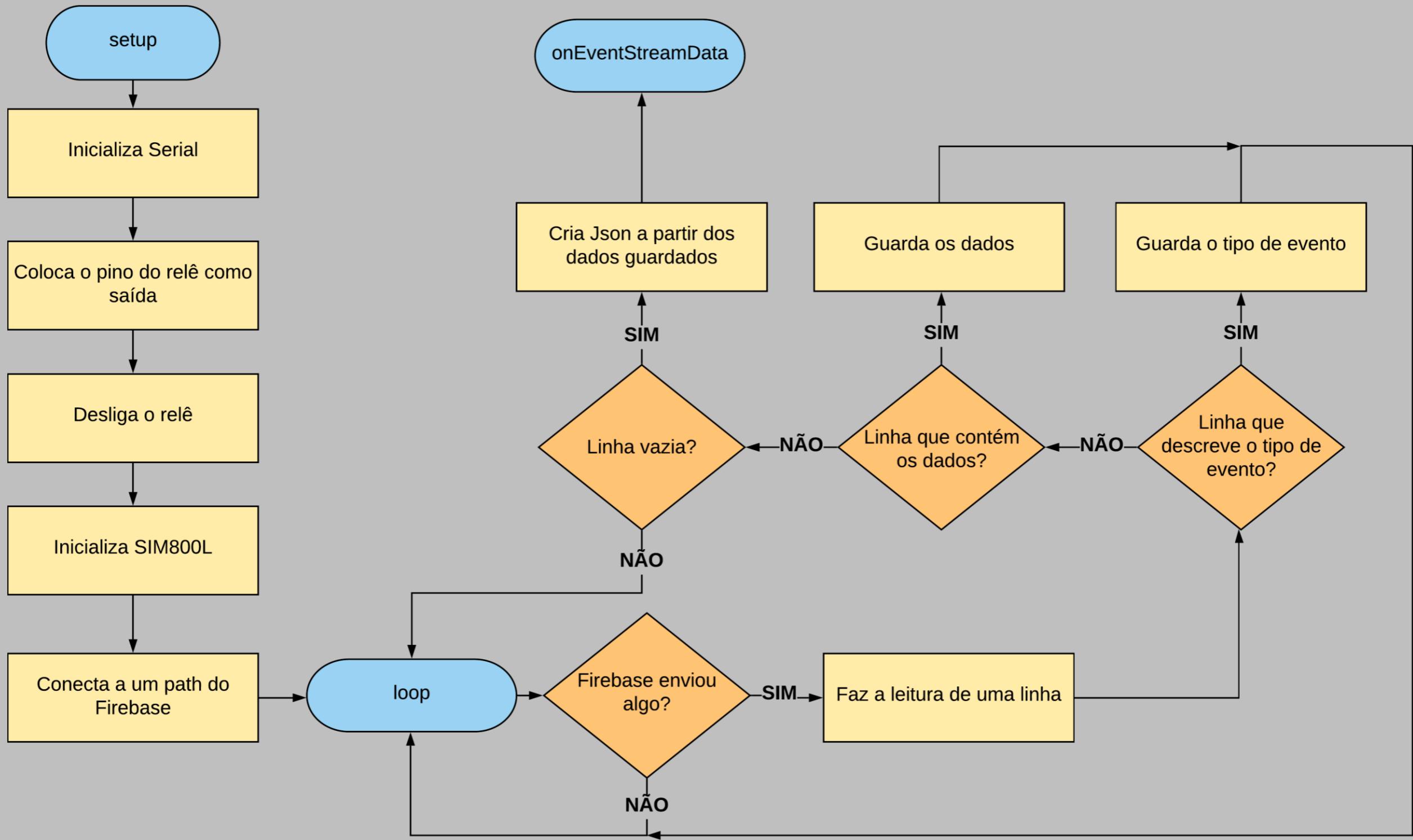
Gerenciador de Placas...

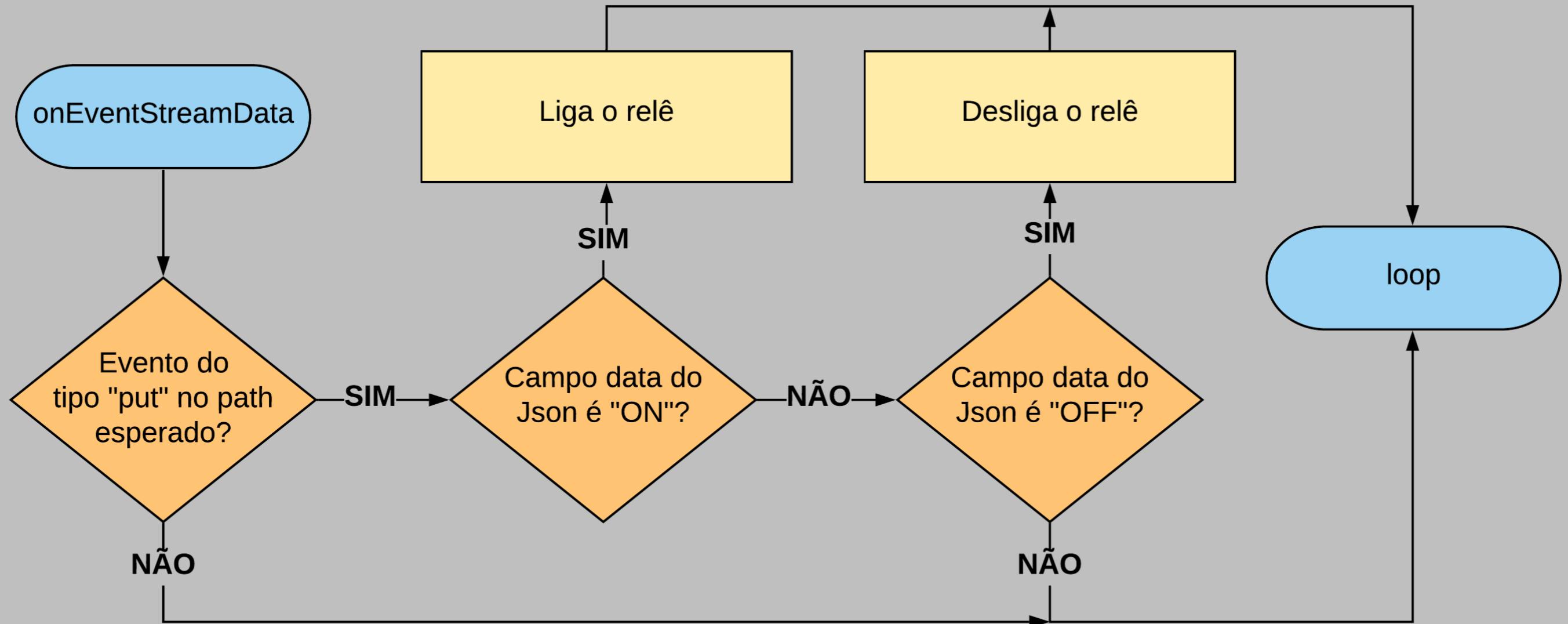
- ▲ Arduino NG or older
- Arduino Robot Control
- Arduino Robot Motor
- Arduino Gemma
- Adafruit Circuit Playground
- Arduino Yún Mini
- Arduino Industrial 101
- Linino One
- Arduino Uno WiFi

- Atmel AVR Xplained minis
- Atmel atmega328p Xplained mini
- Atmel atmega328pb Xplained mini
- Atmel atmega168pb Xplained mini

- ESP32 Arduino
- ESP32 Dev Module
- ESP32 Wrover Module

# **Fluxograma**





# Código

# Código Declarações e variáveis

```
#define TINY_GSM_MODEM_SIM800 //Tipo de modem que estamos usando
#include <TinyGsmClient.h> //https://github.com/vshymanskyy/TinyGSM
#include <ArduinoJson.h> //https://github.com/bblanchon/ArduinoJson
#include <ArduinoHttpClient.h> //https://github.com/arduino-libraries/ArduinoHttpClient

// URL da base de dados fornecido pelo Firebase para a conexão http
#define FIREBASE_HOST "xxxxxxxxxxxx.firebaseio.com"
// Autenticação (no caso não estamos usando)
#define FIREBASE_AUTH ""

//Pinos SIM800L
#define MODEM_RST 5
#define MODEM_PWRKEY 4
#define MODEM_POWER_ON 23
#define MODEM_TX 27
#define MODEM_RX 26

//Pino Relê
#define RELAY_PIN 33
```

# Código Declarações e variáveis

```
//Serial para se comunicar com o SIM800L
HardwareSerial SerialGSM(1);

//Objeto que controla o SIM800L
TinyGsm modemGSM(SerialGSM);

//Socket utilizado no cliente http
TinyGsmClientSecure gsmClient(modemGSM);

//Objeto que fará a comunicação com o Firebase
HttpClient httpClient(gsmClient, FIREBASE_HOST, 443);

//Variáveis para guardar qual o evento
//e os dados que chegarem do Firebase
String event;
String data;
```

# Código setup

```
void setup()
{
    //Inicializa a serial que usaremos no monitor serial
    Serial.begin(115200);

    //Coloca o pino do relê como saída
    pinMode(RELAY_PIN, OUTPUT);

    //Desliga o relê. Pela configuração do nosso relê: HIGH desliga e LOW liga
    digitalWrite(RELAY_PIN, HIGH);

    //Configura o SIM800L
    setupSIM800L();

    //Espera 2 segundos
    delay(2000);

    //Configura a conexão com o Firebase
    setupFirebaseEventStream();
}
```

# Código *setupSIM800L* (1 de 3)

```
void setupSIM800L()
{
    Serial.println("Setup SIM800L...");

    //Reseta o SIM800L
    pinMode(MODEM_RST, OUTPUT);
    pinMode(MODEM_PWRKEY, OUTPUT);
    pinMode(MODEM_POWER_ON, OUTPUT);
    digitalWrite(MODEM_RST, HIGH);
    digitalWrite(MODEM_POWER_ON, HIGH);
    digitalWrite(MODEM_PWRKEY, HIGH);
    delay(100);
    digitalWrite(MODEM_PWRKEY, LOW);
    delay(1000);
    digitalWrite(MODEM_PWRKEY, HIGH);
```

## Código *setupSIM800L* (2 de 3)

```
//Inicializa a comunicação serial com o SIM800L
SerialGSM.begin(115200, SERIAL_8N1, MODEM_RX, MODEM_TX);
delay(3000);

//Mostra informações sobre o SIM800L
Serial.println(modemGSM.getModemInfo());

//Conecta a rede telefônica
Serial.print("Waiting for network...");
if (!modemGSM.waitForNetwork())
{
    Serial.println(" fail");
    delay(10000);
    return;
}
```

# Código *setupSIM800L* (3 de 3)

```
Serial.println(" success");

//Conecta à rede de dados
Serial.print("Connecting gprs ");
if (!modemGSM.gprsConnect("", "", ""))
{
    Serial.println(" fail");
    delay(10000);
    return;
}

Serial.println(" success");
Serial.println("Setup SIM800L Success");
} //setupSIM800L
```

# Código *setupFirebaseEventStream* (1 de 2)

```
void setupFirebaseEventStream()
{
    //Tempo máximo de espera para a requisição
    httpClient.setTimeout(1000);

    //Inicializa a requisição
    httpClient.beginRequest();

    //Path cujos dados iremos receber em tempo real
    String path = "/relayStatus.json";

    //Se está utilizando autenticação
    if(strcmp(FIREBASE_AUTH, "") != 0)
    {
        path += "?auth=" + String(FIREBASE_AUTH);
    }

    //Faz uma requisição HTTP do tipo GET no path
    httpClient.get(path);
```

# Código *setupFirebaseEventStream* (2 de 2)

```
//Informamos ao servidor que queremos os dados via "event-stream", ou seja, em tempo real
httpClient.sendHeader("Accept", "text/event-stream");

//Finaliza a requisição
httpClient.endRequest();

//Verifica o código de resposta da requisição
int httpCode = httpClient.responseStatusCode();

//Se não for 200 significa que houve algum erro
if (httpCode != 200)
{
    Serial.println("Firebase event stream error. HTTP status code: " + String(httpCode));
}
else
{
    Serial.println("Connected to Firebase stream!!!!");
}

Serial.println("Setup Firebase complete");
} //setupFirebaseEventStream
```

# Código *loop*

```
void loop()
{
    //Se o servidor quer nos enviar algo
    if (httpClient.available())
    {
        //Fazemos a leitura
        readFirebaseEventStreamLine();
    }
}
```

# Código *readFirebaseEventStreamLine* (1 de 2)

```
//Faz a leitura dos dados enviados pelo Firebase pela conexão que fizemos
//Sempre que algo for alterado no "path" que fizemos a conexão,
//o Firebase vai nos enviar o seguinte:
//event: 'tipo de evento'\r\n\data: 'dados que foram alterados'\r\n\r\n
//Então vamos ler linha a linha (ou seja até cada \n) e no final
//teremos uma linha vazia ('\r\n' depois do '\r\n'do campo "data")
//Após a linha vazia já temos o tipo de evento e os dados salvos
void readFirebaseEventStreamLine()
{
    //Lemos uma linha
    String line = httpClient.readStringUntil('\n');

    //Se começar com "event:"
    if(line.startsWith("event:")){
        //Guardamos a informação de qual tipo de evento
        line.replace("event:", "");
        line.trim();
        event = line;
    }
}
```

# Código *readFirebaseEventStreamLine* (2 de 2)

```
//Se começar com "data:"  
else if(line.startsWith("data:"))  
{  
    //Guardamos os dados  
    line.replace("data:", "");  
    line.trim();  
    data = line;  
}  
//Se a linha tiver tamanho 0 significa que já lemos toda a resposta  
else if(line.length() == 0 && data.length() != 0)  
{  
    //Criamos um objeto json com os dados  
    StaticJsonBuffer<1024> jsonBuffer;  
    JsonObject &root = jsonBuffer.parseObject(data);  
  
    //Se o json é válido  
    if (root.success())  
    {  
        //Chamamos a função que irá manipular os dados  
        onEventStreamData(event, root["path"].asString(), root["data"].asString());  
    }  
}  
} //readFirebaseEventStreamLine
```

# Código *onEventStreamData*

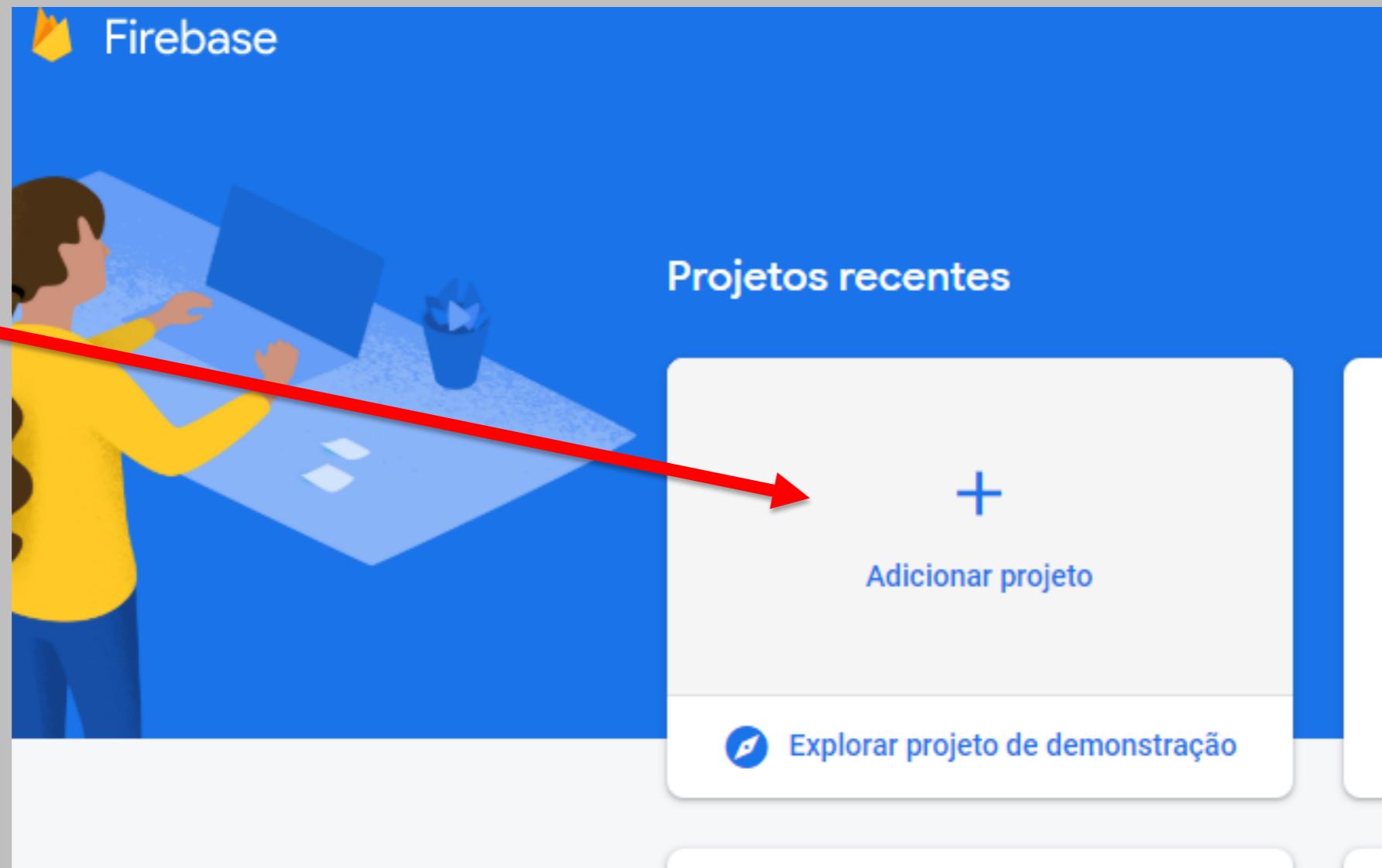
```
//Função chamada toda vez que recebemos algo do firebase
//event: tipo de evento
//path: path relativo ao que estamos conectados, ou seja,
//se nos conectamos ao /relayStatus.json um "/" aqui significa que a raiz /relayStatus foi alterada
//value: valor que foi enviado no "path"
void onEventStreamData(String event, String path, String value)
{
    //Verifica se é um evento do tipo "put" ocorreu em /relayStatus
    if(event == "put" && path == "/") {
        value.toUpperCase(); //Coloca tudo em maiúsculo já que independe a maneira que foi enviada

        if(value == "ON") { //Se o valor for "ON"
            digitalWrite(RELAY_PIN, LOW); //Ligamos o relê
        }
        else if(value == "OFF") { //Se o valor for "OFF"
            digitalWrite(RELAY_PIN, HIGH); //Desligamos o relê
        }
    }
}
```

# Firebase

**Firebase** <https://console.firebaseio.google.com/>

Clique em  
“Adicionar  
projeto”



# Firebase

Digite o nome  
do projeto e  
clique em  
“Continuar”

X Criar um projeto(Passo 1 de 3)

Vamos começar com um  
nome para o projeto?

Nome do projeto

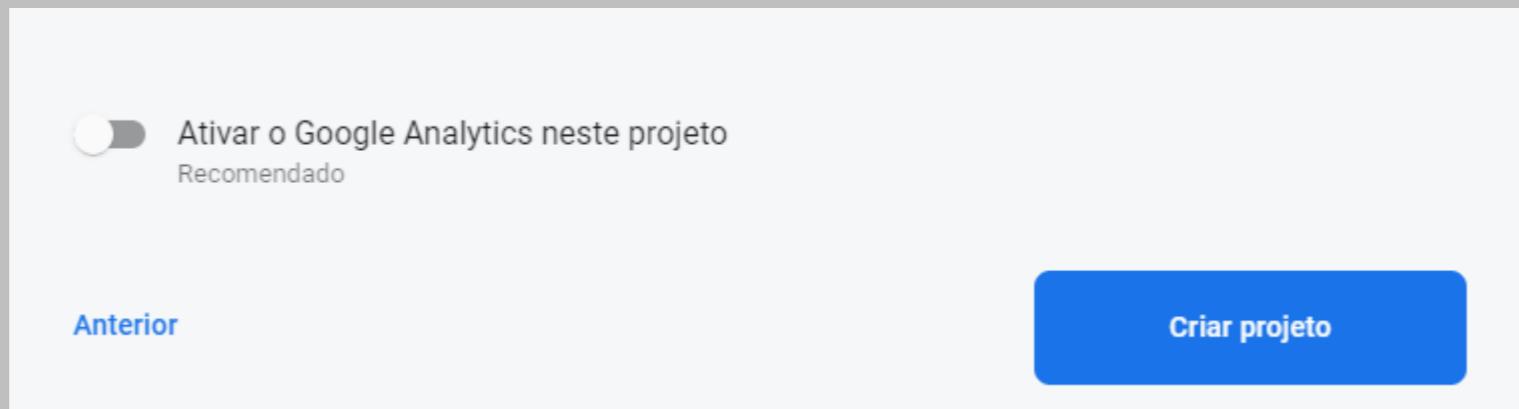
Meu Projeto

 meu-projeto [REDACTED]

Continuar

# Firebase

Desative a  
opção Analytics  
já que não  
vamos utilizar  
neste exemplo.  
Clique em  
“Criar Projeto”



# Firebase

Clique em  
“Database”

The image shows a screenshot of the Firebase console interface. On the left, there is a sidebar with the title "Desenvolver" and a list of services: Authentication, Database, Storage, Hosting, Functions, and Machine Learning. A red arrow points from the text "Clique em ‘Database’" to the "Database" option in the sidebar. The main content area is titled "Meu Projeto" and features a large blue banner with the text "Comece adicionando Firebase ao seu aplicativo". At the top right of the main area, it says "Plano Spark".

# Firebase

**Na seção Realtime Database**

The screenshot shows the Firebase console's database selection screen. At the top, there is a button with a green smiley face icon and the text "pode ser útil para mim?" followed by "Saiba mais". Below this, there are two main options: "Cloud Firestore" and "Realtime Database". "Cloud Firestore" is highlighted with an orange bar at the bottom of its card. The "Realtime Database" card features a yellow background with three dark grey rectangular icons containing white symbols. To the right of the card, the text "Realtime Database" is displayed, followed by a description: "Banco de dados original do Firebase. Assim como o Cloud Firestore, ele é compatível com a sincronização de dados em tempo real." Below this text are two links: "Ver os documentos" and "Saiba mais". At the bottom right of the card is a blue button with the text "Criar banco de dados". A red arrow points from the text "Na seção Realtime Database" down to the "Realtime Database" card. Another red arrow points from the text "Clique em ‘Criar banco de dados’" down to the "Criar banco de dados" button.

**Clique em “Criar banco de dados”**

# Firebase

## Regras de segurança para o Realtime Database

X

Depois de definir sua estrutura de dados, você precisará gravar regras para proteger seus dados.

[Saiba mais ↗](#)

### Selecione modo teste

Iniciar no modo bloqueado

Para tornar seu banco de dados privado, basta negar todas as leituras e gravações

Iniciar no modo de teste

Configure rapidamente ao permitir todas as leituras e gravações no seu banco de dados

```
{  
  "rules": {  
    ".read": true,  
    ".write": true  
  }  
}
```

! Qualquer pessoa com sua referência de banco de dados poderá fazer leituras ou gravações no seu banco de dados

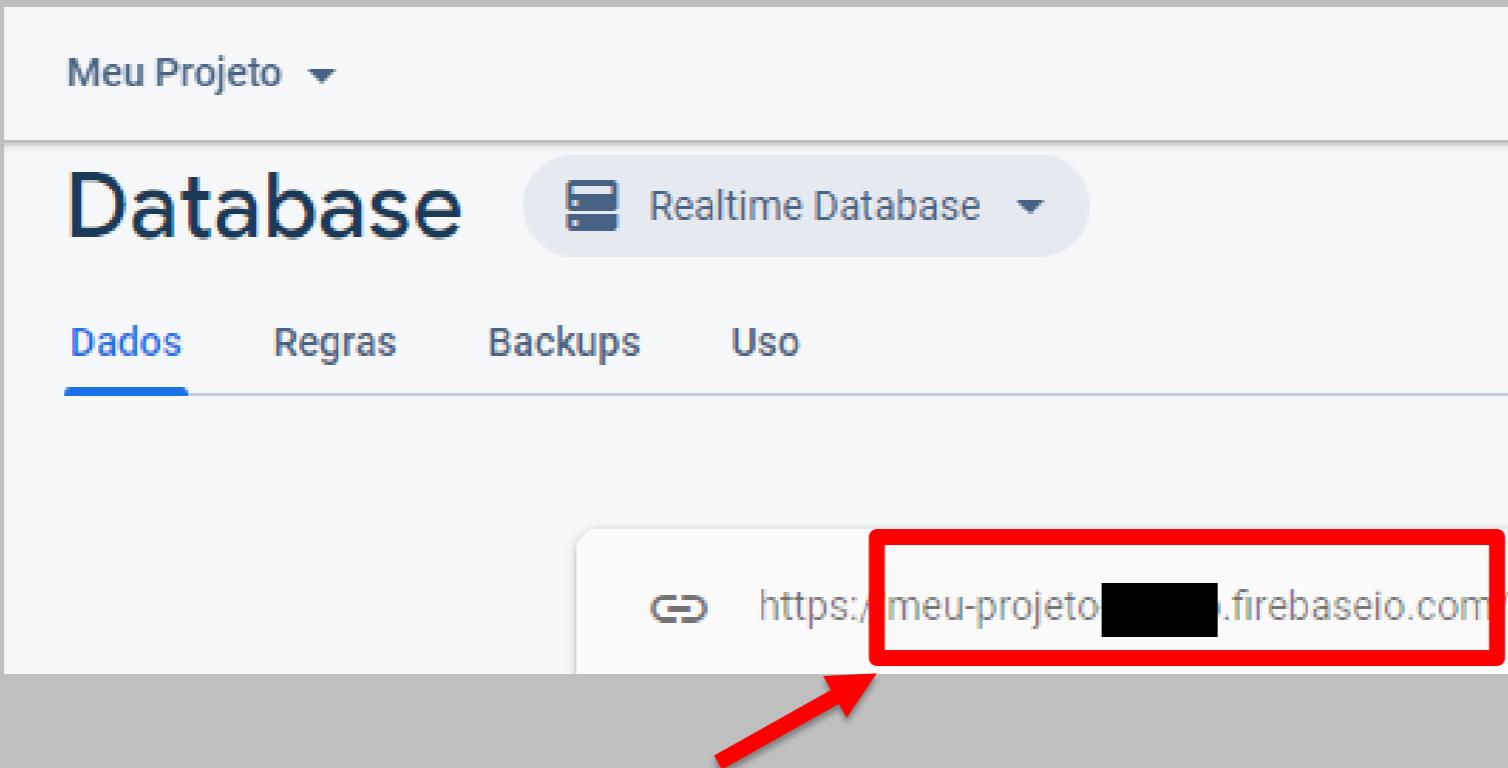
Cancelar

Ativar



Clique em “Ativar”

# Firebase



Esta é a url que se coloca em **FIREBASE\_HOST** no código fonte (sem https:// no .ino) e também no **databaseURL** do app Fernando K (com https://)

**App Fernando K**

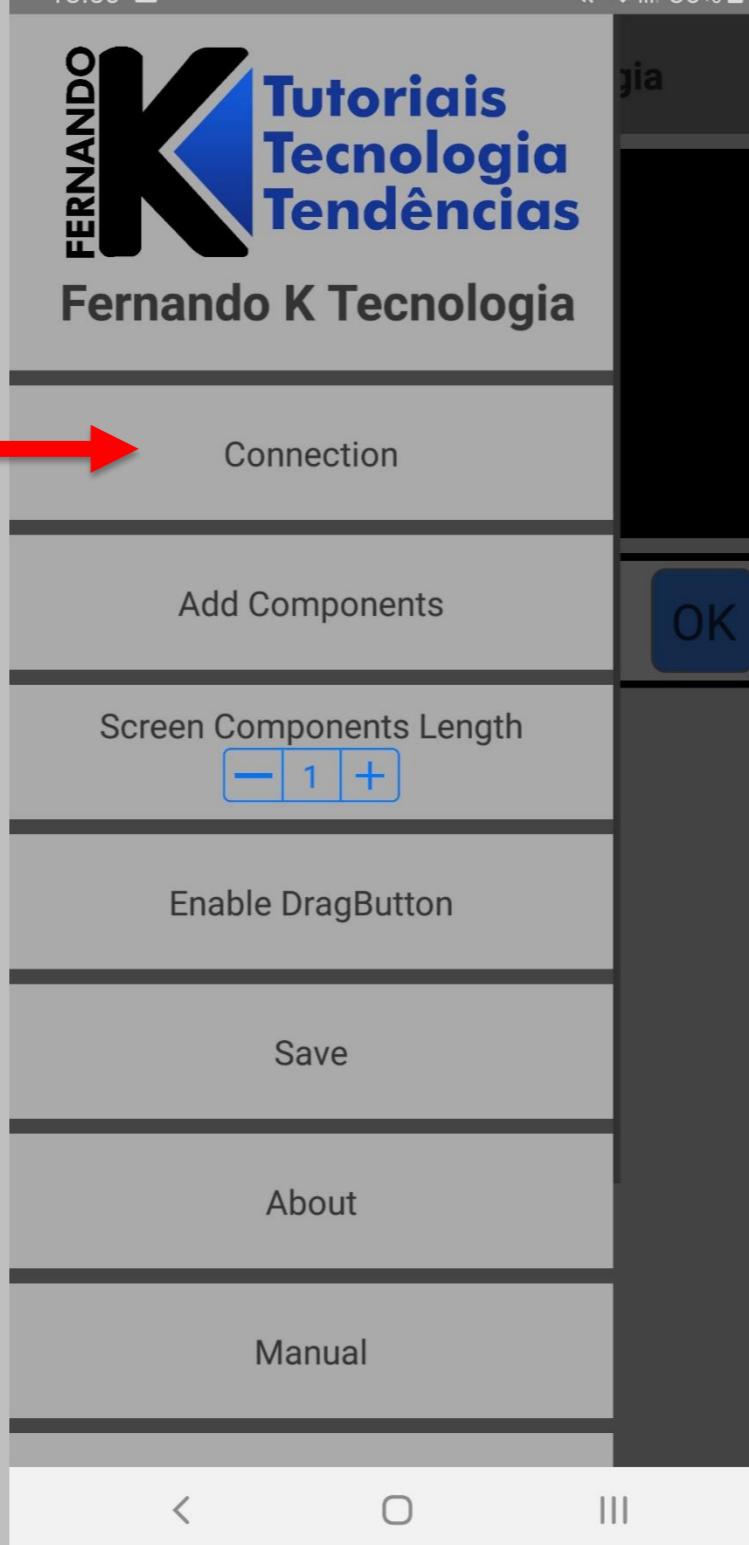
# AppFernandoK

Clique Aqui →

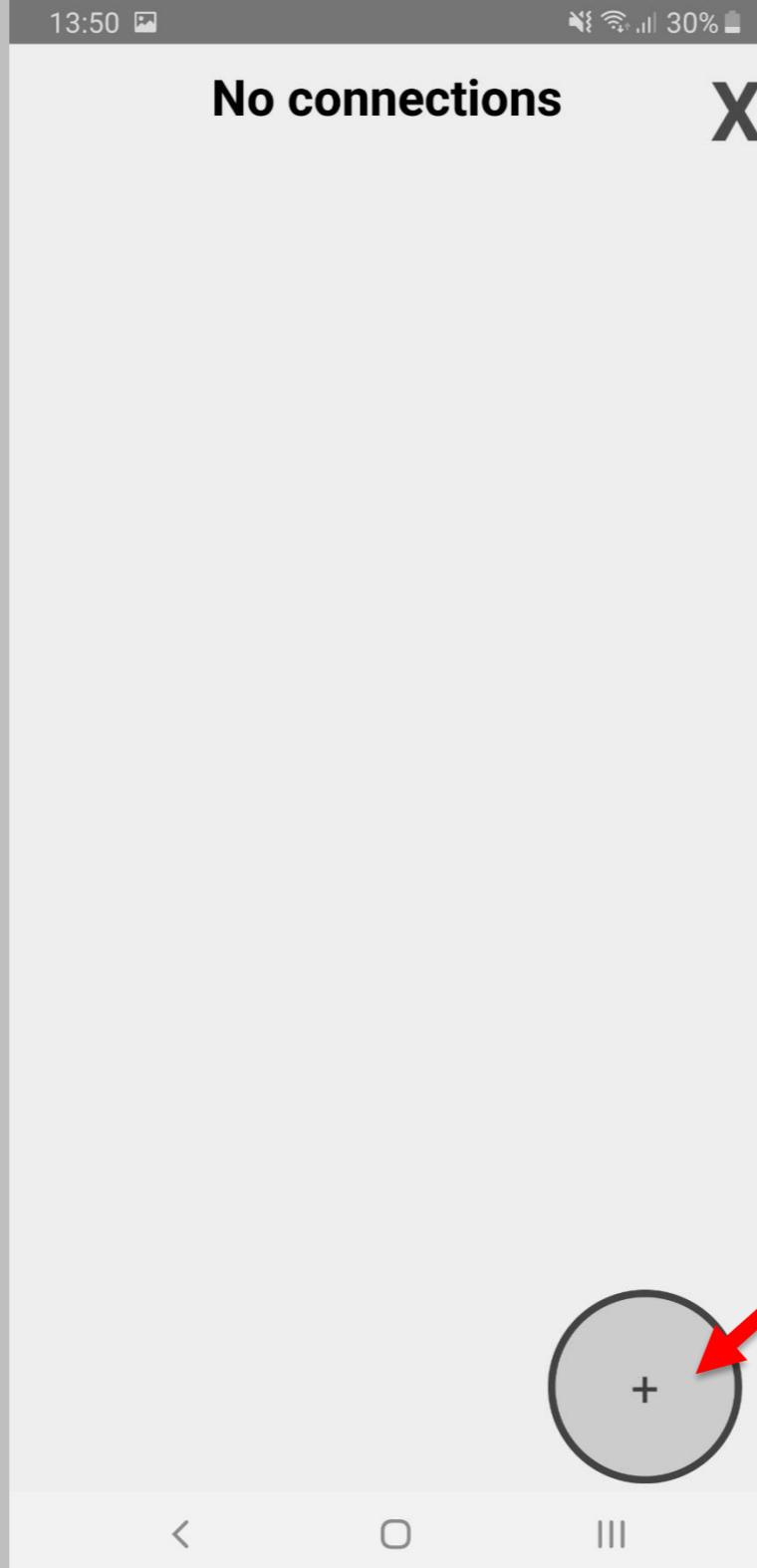


# AppFernandoK

Clique em  
“Connection”



# AppFernandoK

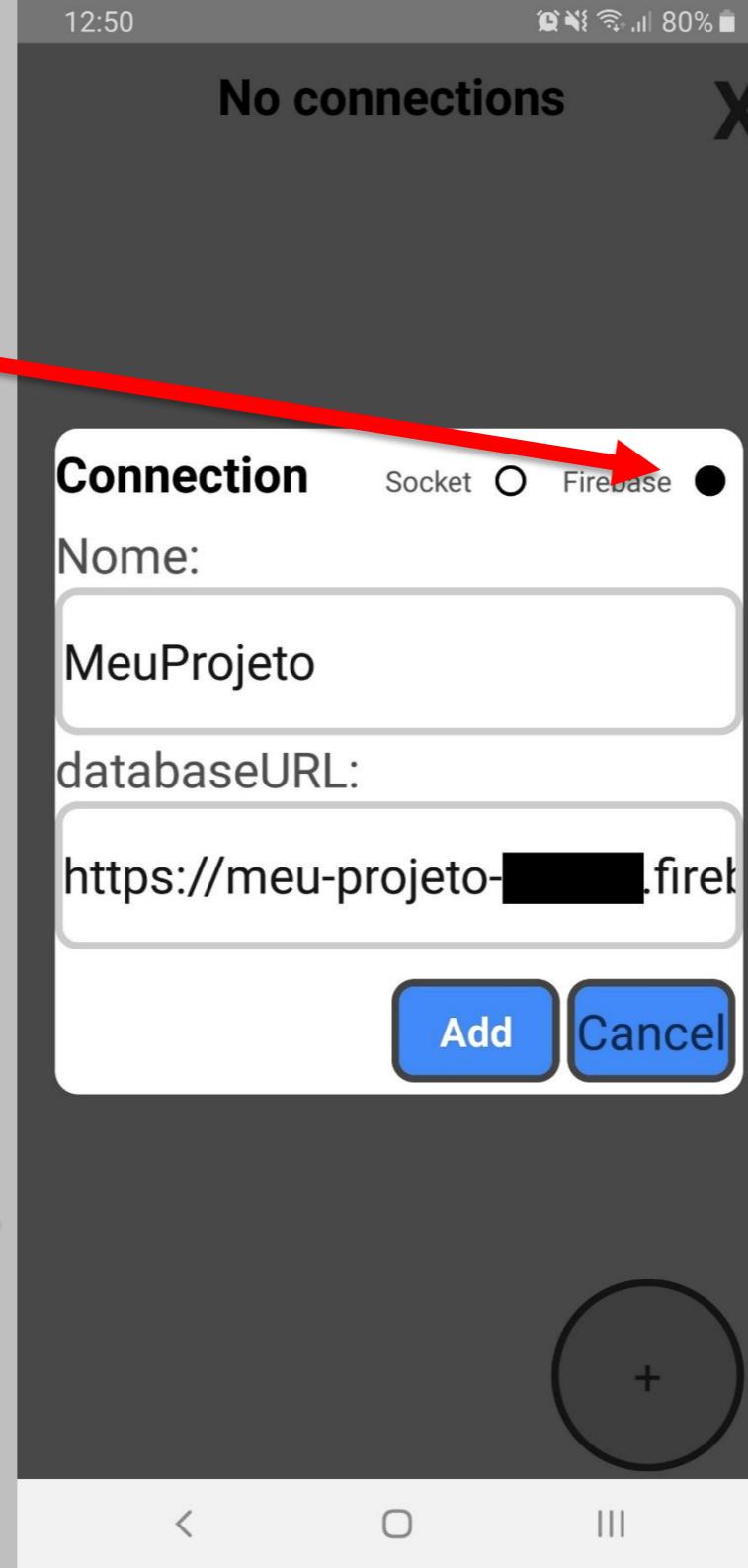


Clique aqui

# AppFernandoK

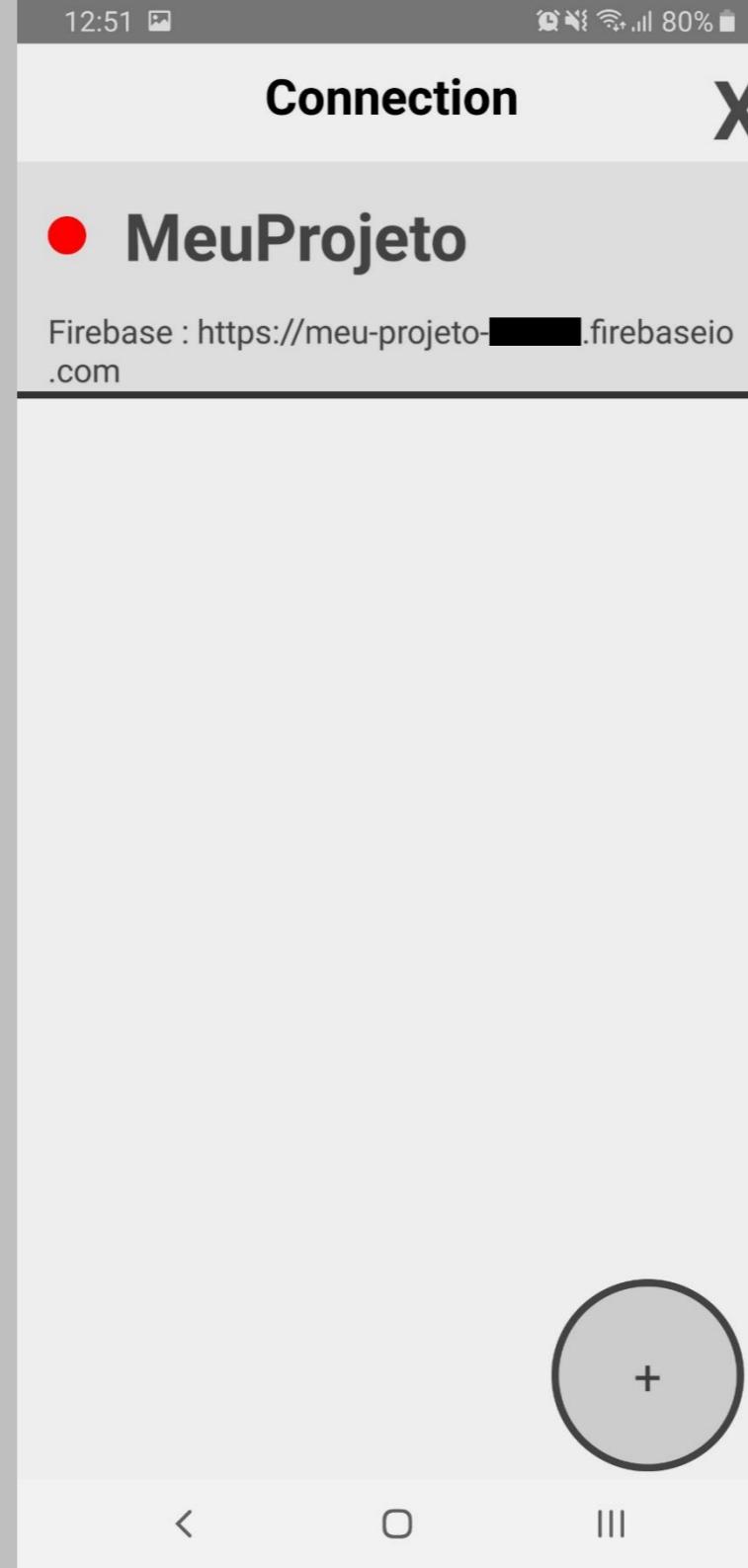
Clique aqui

Adicione nome  
para a conexão e  
adicone  
dataBaseURL  
(o mesmo em  
**FIREBASE\_HOST**  
do .ino) com https://  
e clique em “Add”



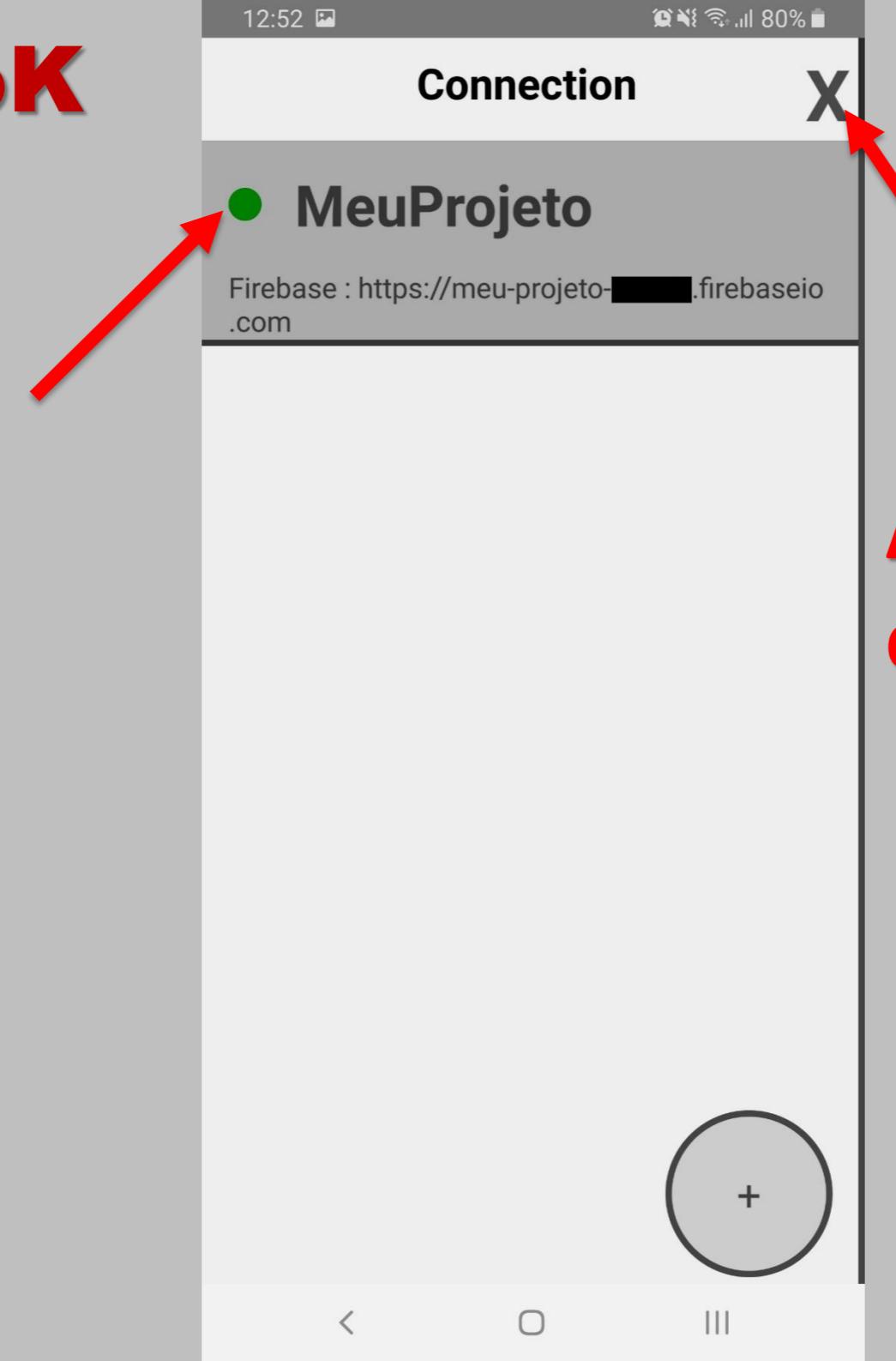
# AppFernandoK

A conexão aparecerá  
na lista. Clique nela.



# AppFernandoK

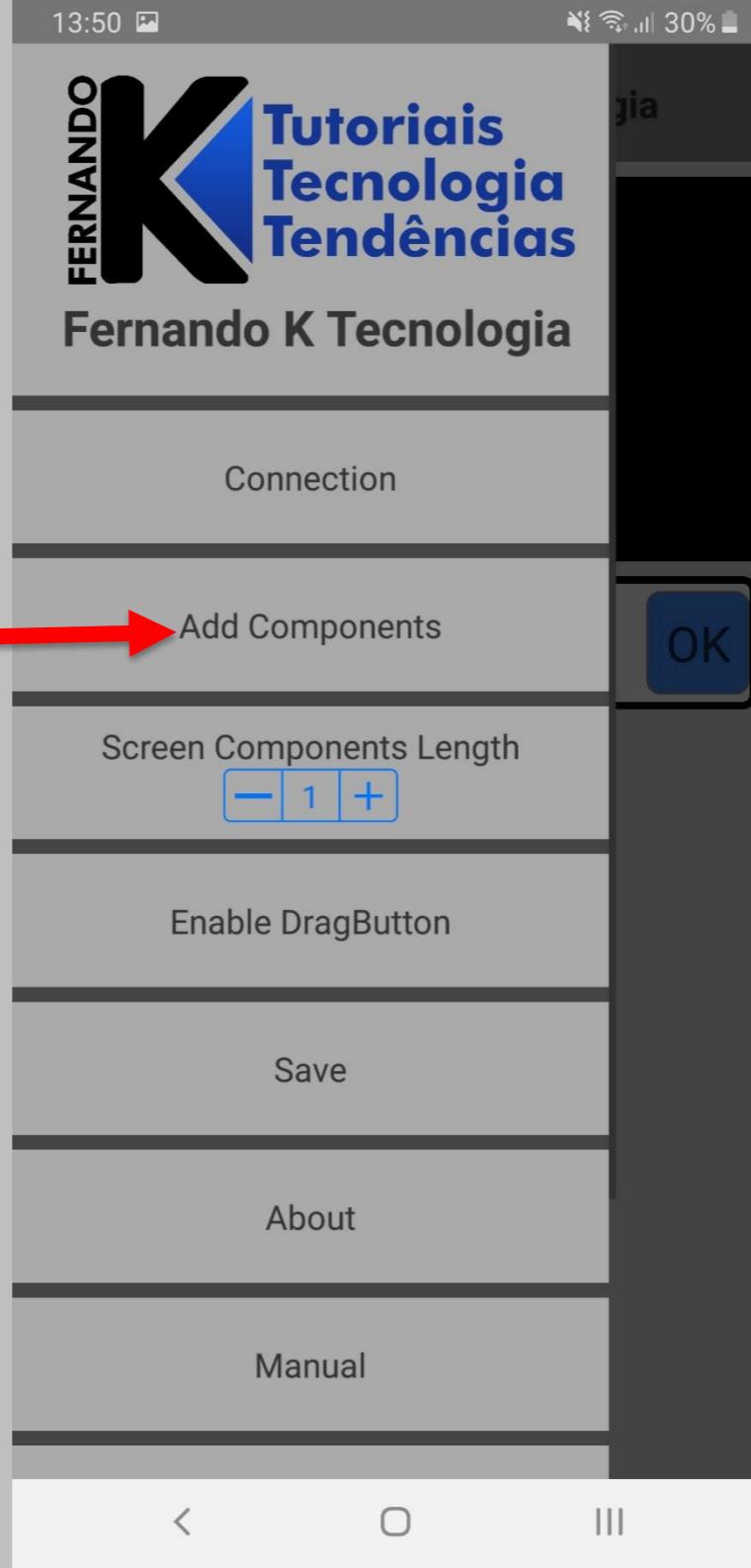
Se tudo ocorreu  
bem o ícone  
ficará verde



Agora feche a lista  
clicando aqui

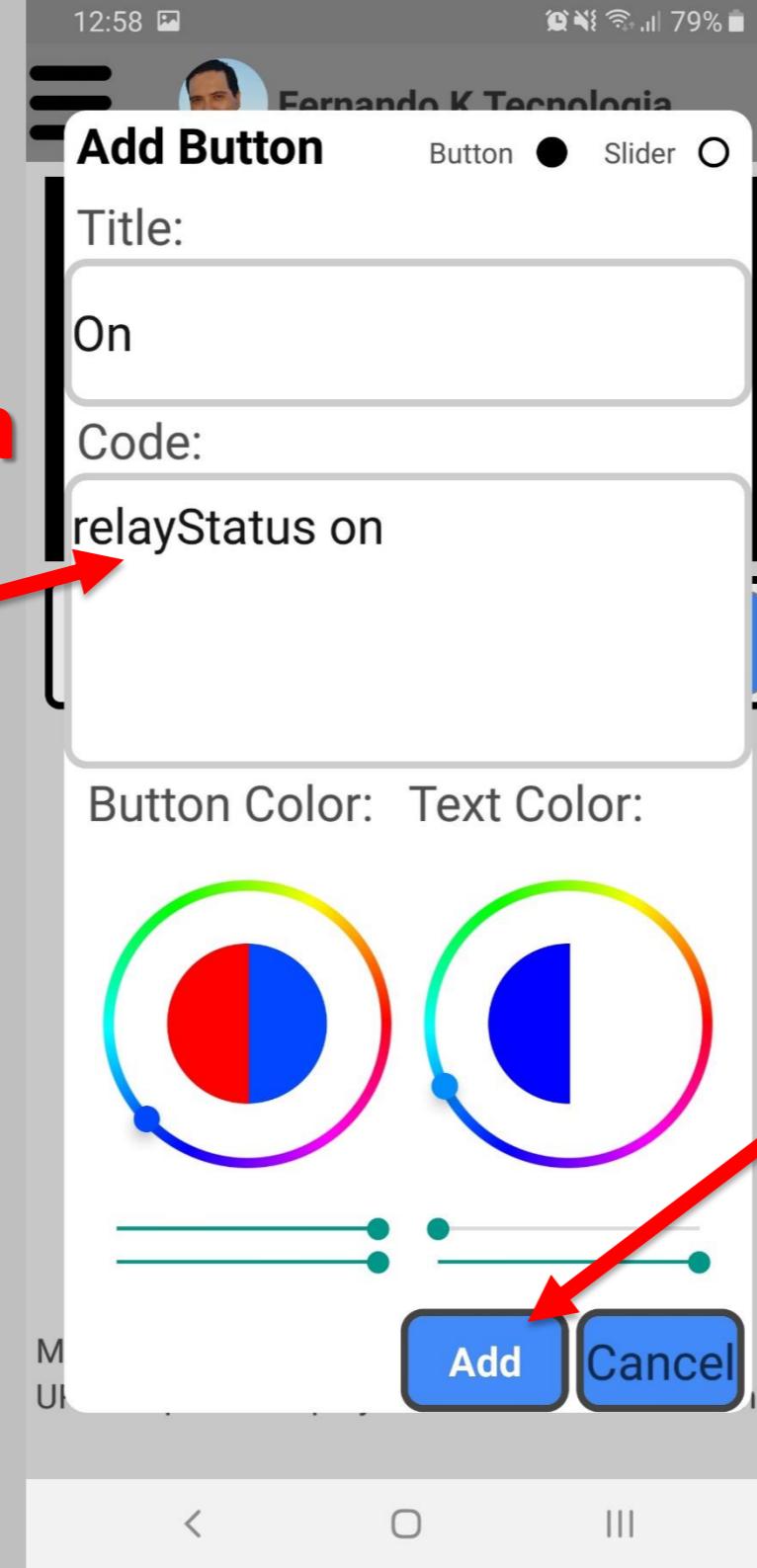
# AppFernandoK

Clique em  
“Add Components”



# AppFernandoK

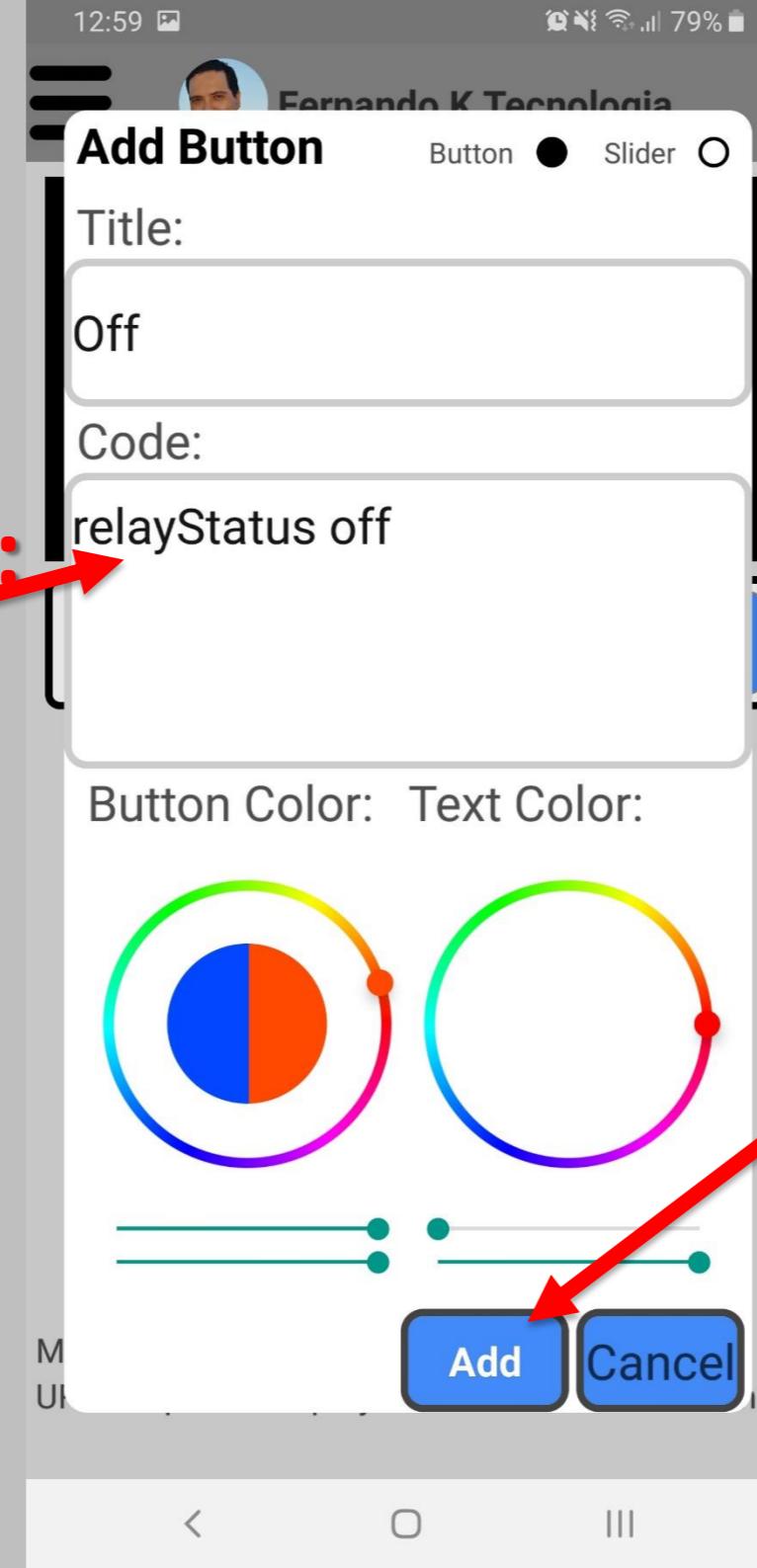
Adicione um botão  
que irá ligar o relê com  
o seguinte código:  
**relayStatus on**



Clique em “Add”

# AppFernandoK

Adicione um botão  
que irá desligar o relê  
com o seguinte código:  
**relayStatus off**

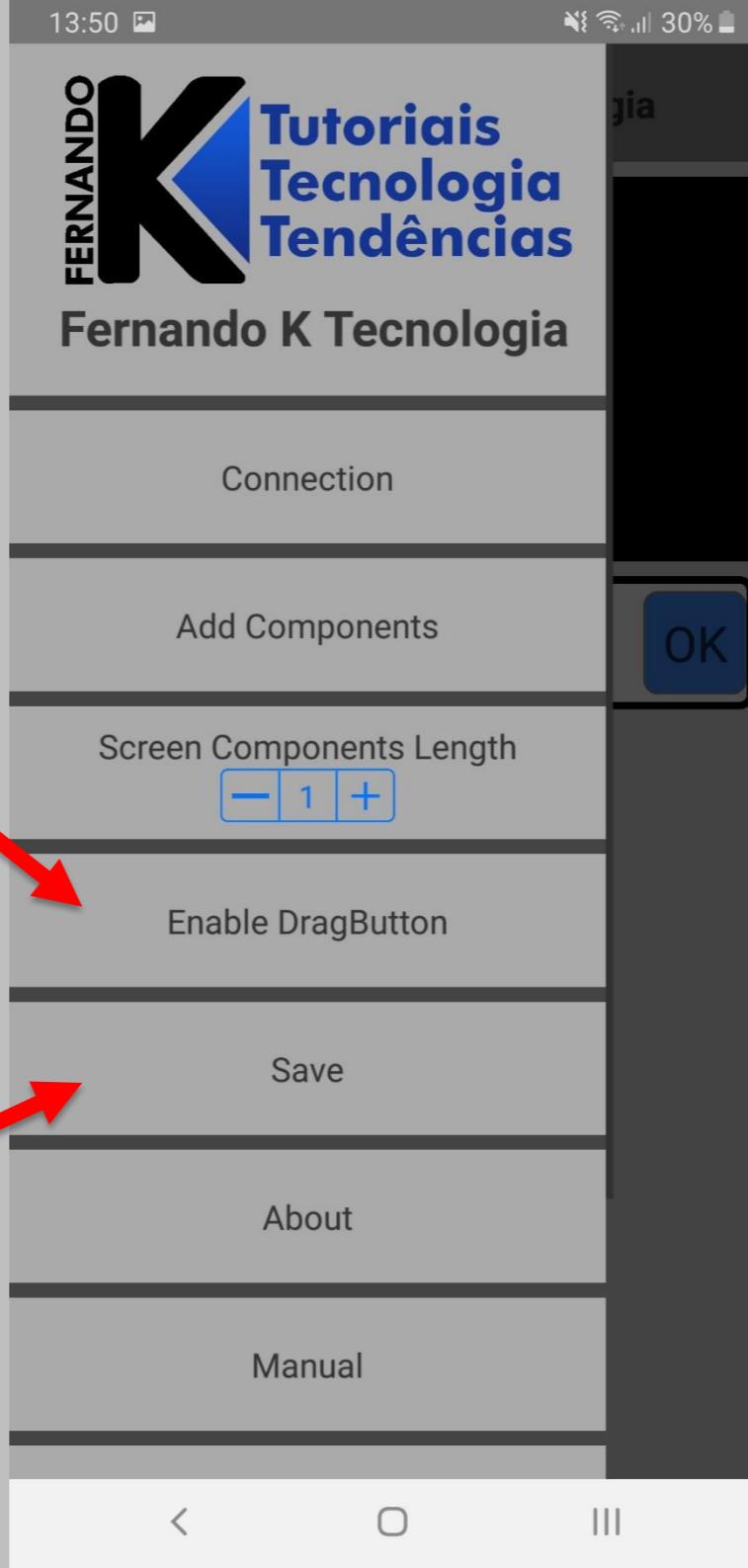


Clique em “Add”

# AppFernandoK

Você pode  
mover os  
botões  
clicando  
aqui

**IMPORTANTE!!!**  
Lembre-se de  
salvar tudo o  
que fez



# AppFernandoK

**Agora basta clicar  
no botão “on”  
Quando quiser ligar  
o relê e no botão  
“off” quando quiser  
desligar**



**Em [www.fernandok.com](http://www.fernandok.com)**

**Download arquivos PDF e INO do código fonte**

