

DevTitans

Telegram

Telegram

- Telegram é um serviço de mensagens instantâneas baseado em nuvem, multiplataforma e **freemium**.
- **Freemium** é a junção dos termos **free**, ou gratuito, e **premium**, que carrega um serviço de qualidade e com certo grau de exclusividade
- O serviço também oferece **outras possibilidades** como chamadas de vídeo criptografadas de ponta a ponta, VoIP, compartilhamento de arquivos e vários outros recursos
- Mas o que o diferencia de outras plataformas de mensagens é a **capacidade de criar bots**

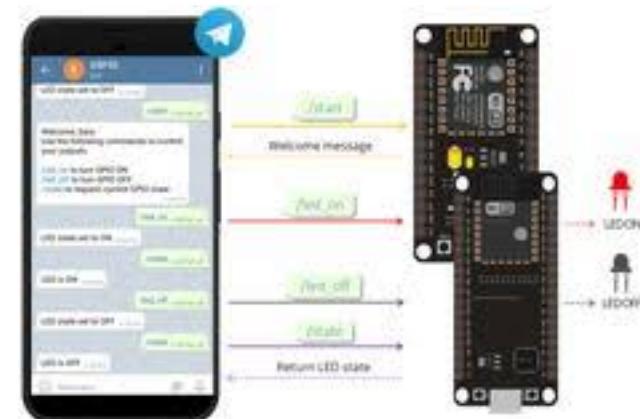


Bots

- **Bot** é uma abreviação do Inglês **robot (robô)**
- Os **bots** podem ser entendidos como programas, softwares ou aplicações de computador, que são desenvolvidos para executar tarefas **específicas e repetitivas** na internet, de forma automatizada
- Os bots são aplicativos de terceiros que são executadas **dentro** do Telegram
- Os usuários podem interagir com bots enviando mensagens, comandos e solicitações em linha
- Você controla seus bots usando solicitações HTTPS para **Telegram Bot API**

Telegram e ESP32

- O ESP32 irá **interagir com o bot do Telegram** para receber e manipular as mensagens e enviar respostas
- Nesta aula aprenderemos como usar bot do Telegram para enviar mensagens ao ESP32 para solicitar leituras de sensores (umidade e temperatura) e alterar o estado de atuadores (LEDs)
- Isso pode ser feito de qualquer lugar
- Você só precisa do Telegram e acesso à internet

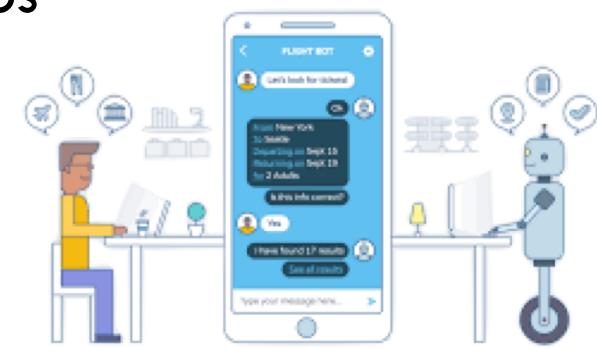


Chatbot

- Nesta aula vamos implementar um chatbot no ESP32 integrado com o Telegram
- Um chatbot é um software que tenta simular uma conversa entre duas pessoas, interagindo com perguntas e respostas e com isso será possível no neste caso controlar atuadores (LEDs) e fazer a leitura de sensores (umidade e temperatura)
- Podemos até mesmo criar alertas, como por exemplo, quando detectar algo na nossa casa, podemos receber diretamente no chat o que está ocorrendo

Chatbot

- Normalmente para controlar nossos dispositivos, ficamos limitados a rede WiFi local ou, se quisermos acessar externamente, temos que fazer uma configuração com um serviço de DNS dinâmico, liberar porta no modem, colocar IP estático e diversas outras coisas, para poder acessar de fora de casa
- Uma das ideias por trás da internet das coisas é que tudo seja o mais fácil possível para os usuários
- No caso dos chatbots, basta você ter o aplicativo correspondente e conversar com seu dispositivo através de comando pré-definidos

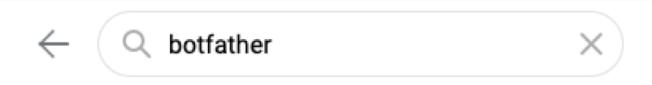


Gerando um bot no Telegram

Passos

- Para conseguir um Token a ser utilizado no código que vai se comunicar com a API do Telegram, siga os passos:
 1. Abra um chat com o **@BotFather** no Telegram e toque em **Iniciar**
 2. Digite o comando **/newbot** para iniciar o processo de criação do bot
 3. Vai ser requisitado o **nome do bot**, então digite o nome de sua preferência
 4. Depois será requisitado o **username do bot**, que basicamente é a forma que você inicia a conversa com ele
 5. Depois disso o **@BotFather** vai devolver pra você um Token de acesso com o formato **NNNNNNNNNN:um-monte-de-letra-e-numero**

Criação do bot no Telegram

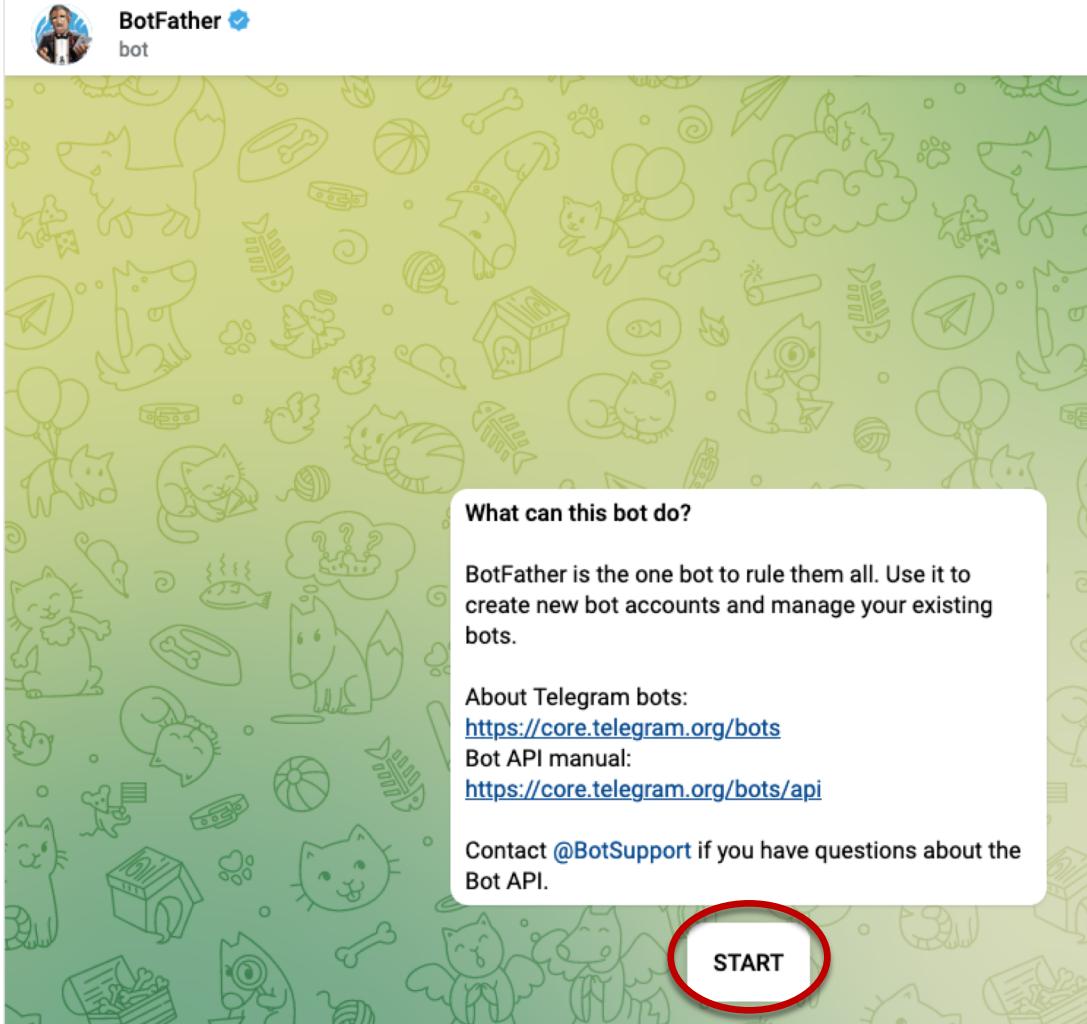


A screenshot of the Telegram search interface. At the top, there is a search bar with the text "botfather" and a magnifying glass icon. Below the search bar is a navigation menu with tabs: "Chats" (which is underlined in blue), "Media", "Links", "Files", "Music", and "Voic". Underneath the search bar, the text "Global search" is followed by a "show more" link. The main area displays a list of search results, each consisting of a small profile picture, the bot's name, and its username. The results are:

- BotFather ✅
@BotFather
- BotFather
@Bott_Father
- BotFather
@father_bot3
- BotFather
@botphather
- TB The BotFather
@BotFather_QBot
- BotFather
@bot_father_createbot
- @
@BotFather
@newbot1

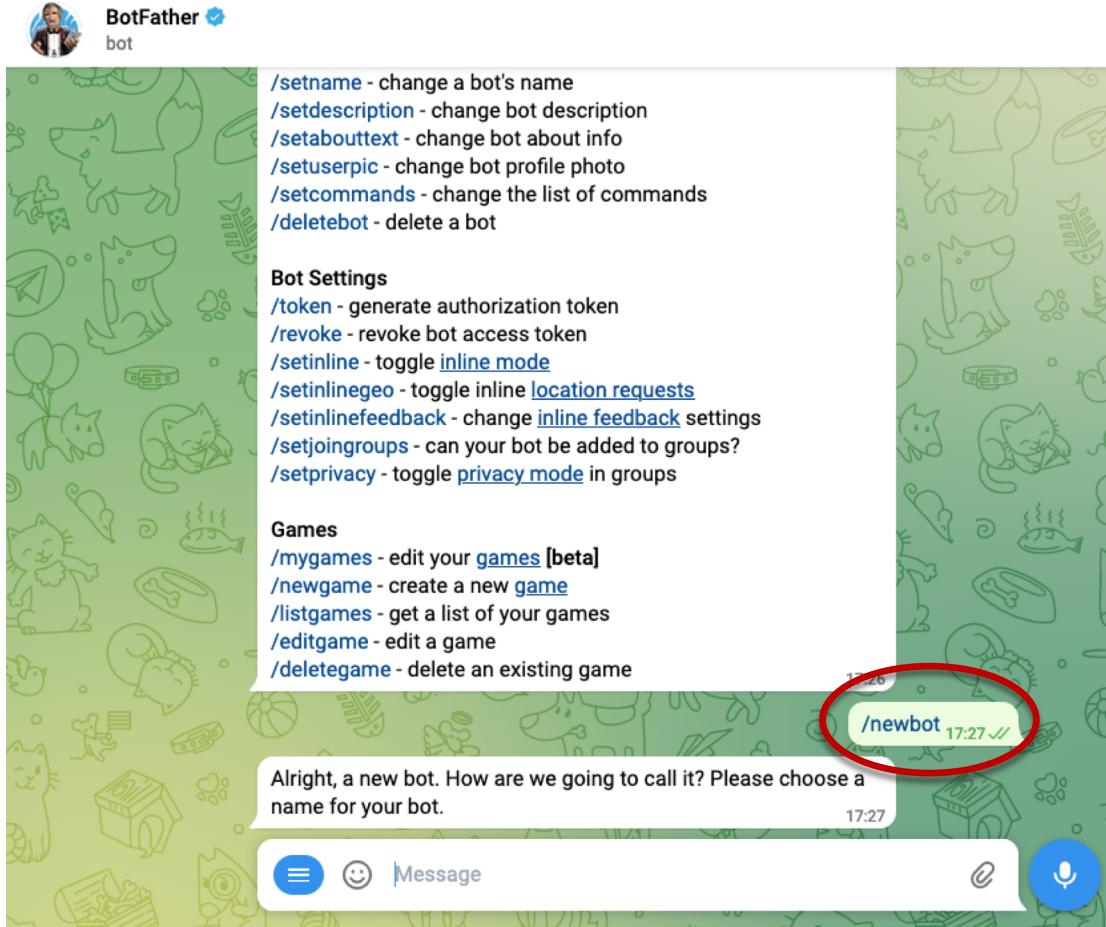
Faça uma busca
por @BotFather no
Telegram

Criação do bot no Telegram



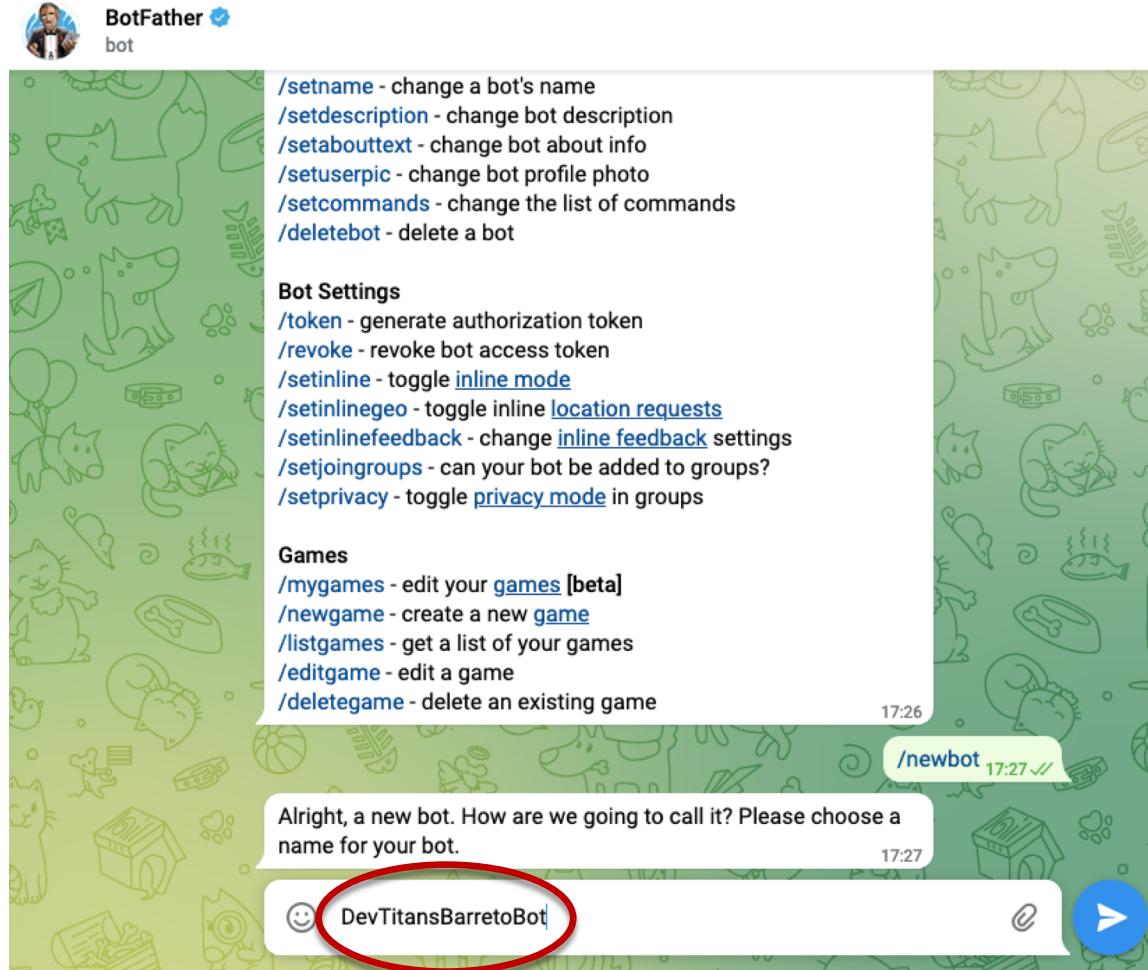
Clique em Start

Criação do bot no Telegram



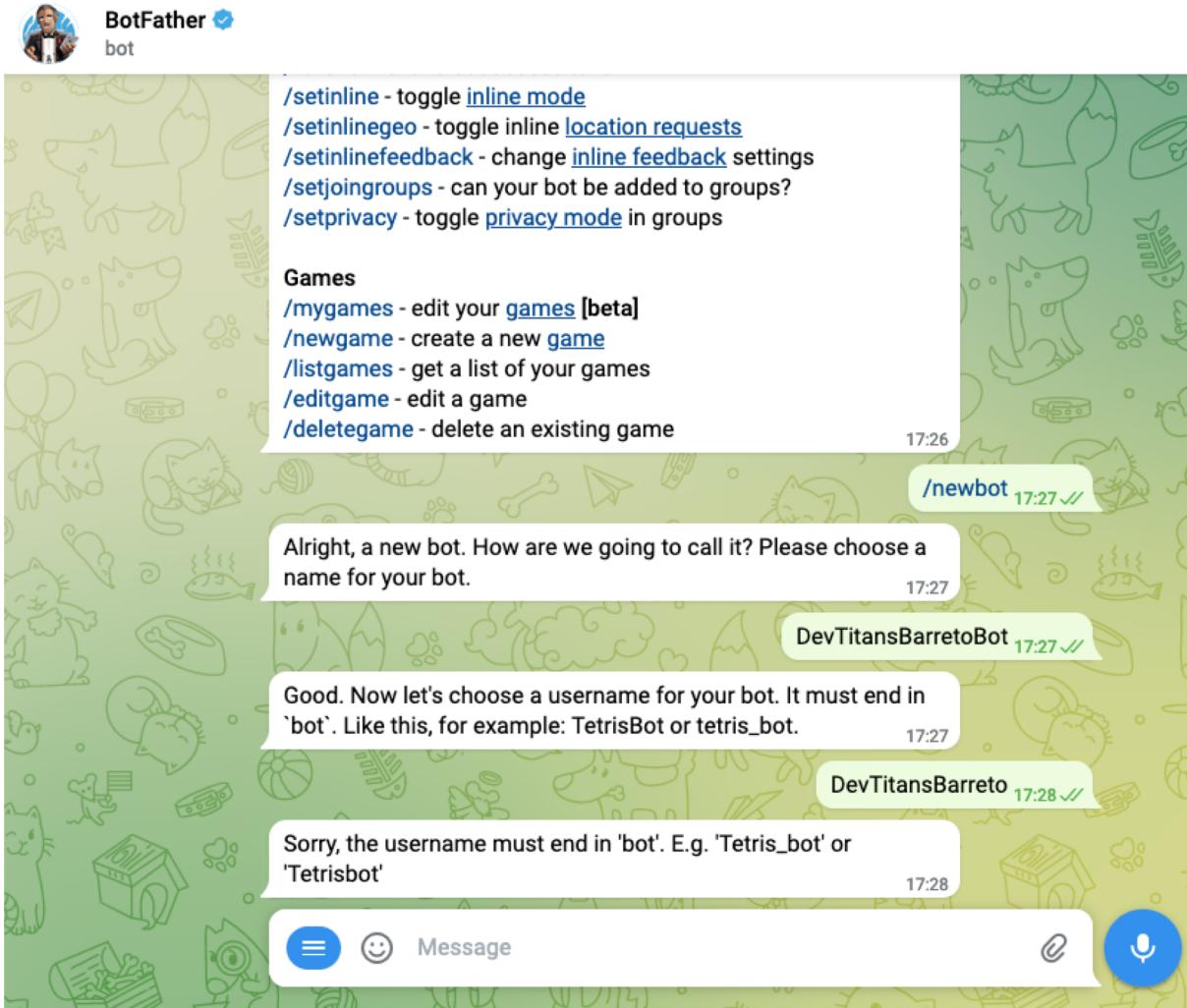
Digite **/newbot**

Criação do bot no Telegram



Digite o nome
do bot (não
precisa ter “bot”
no nome)

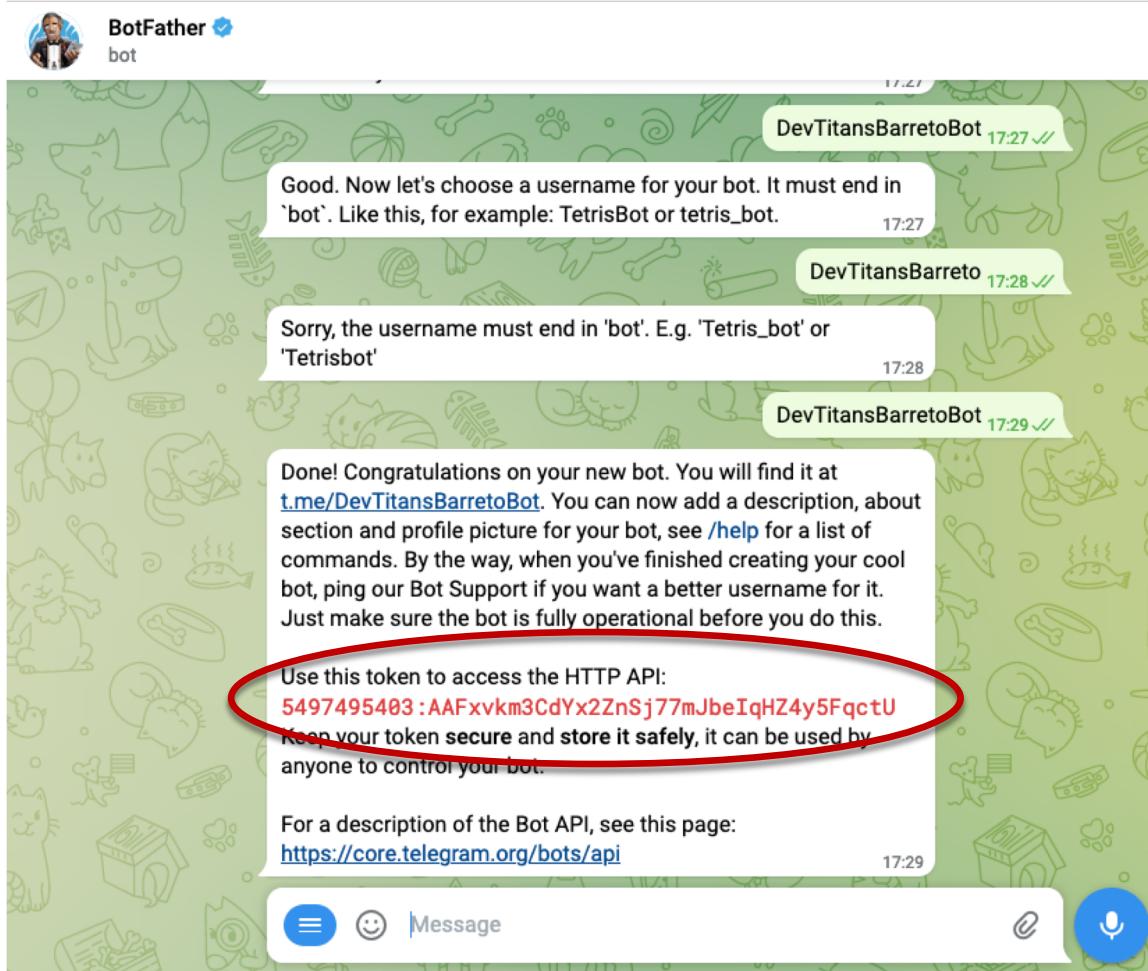
Criação do bot no Telegram



Digite o nome
do usuário
(precisa ter
“bot” no nome)

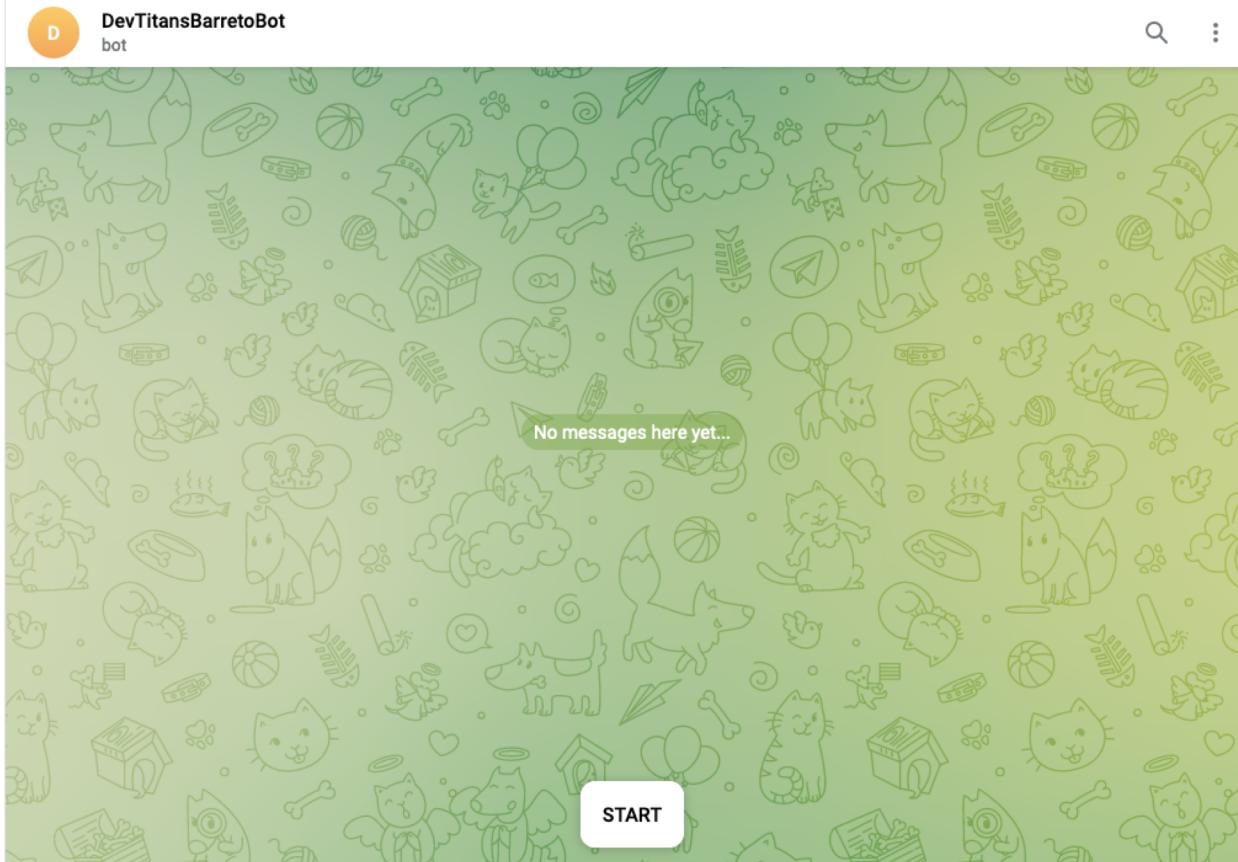
Pode ser o
mesmo nome
do bot

Criação do bot no Telegram



Esse é o token para acesso à API HTTP (necessário no código do ESP32)

Criação do bot no Telegram

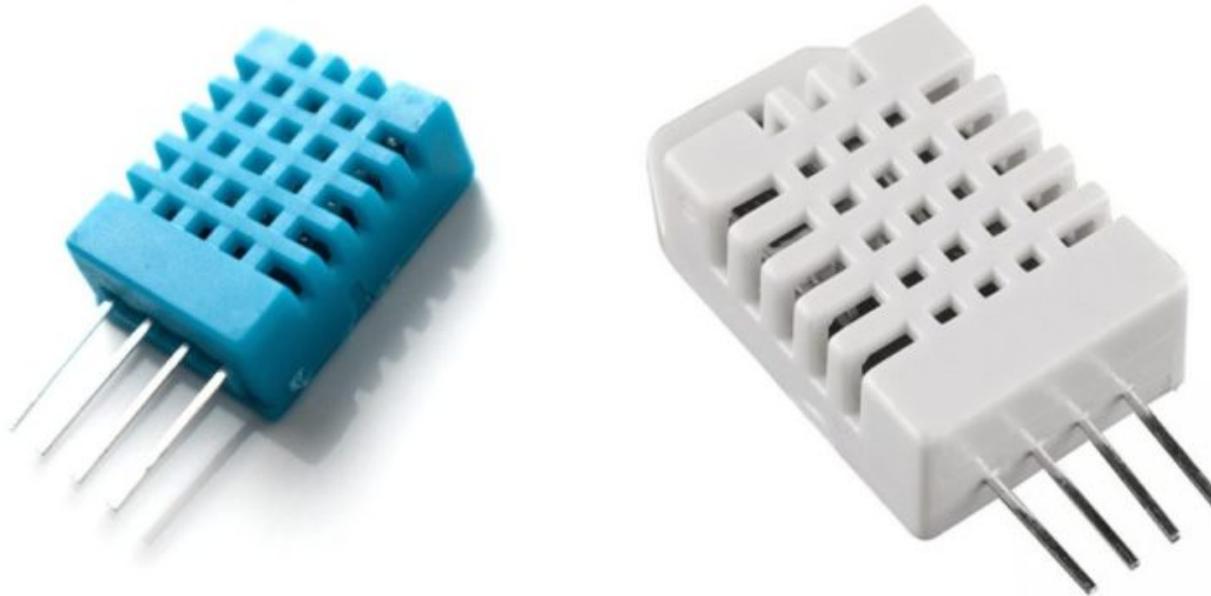


O bot já está
pronto para
interagir com o
ESP32



Hardware e componentes

Sensor DHT11 e DHT22



Utiliza um termistor (resistor sensível a temperatura) e um sensor capacitivo de umidade para medir a temperatura e a umidade do ar ambiente

Sensor DHT11 e DHT22

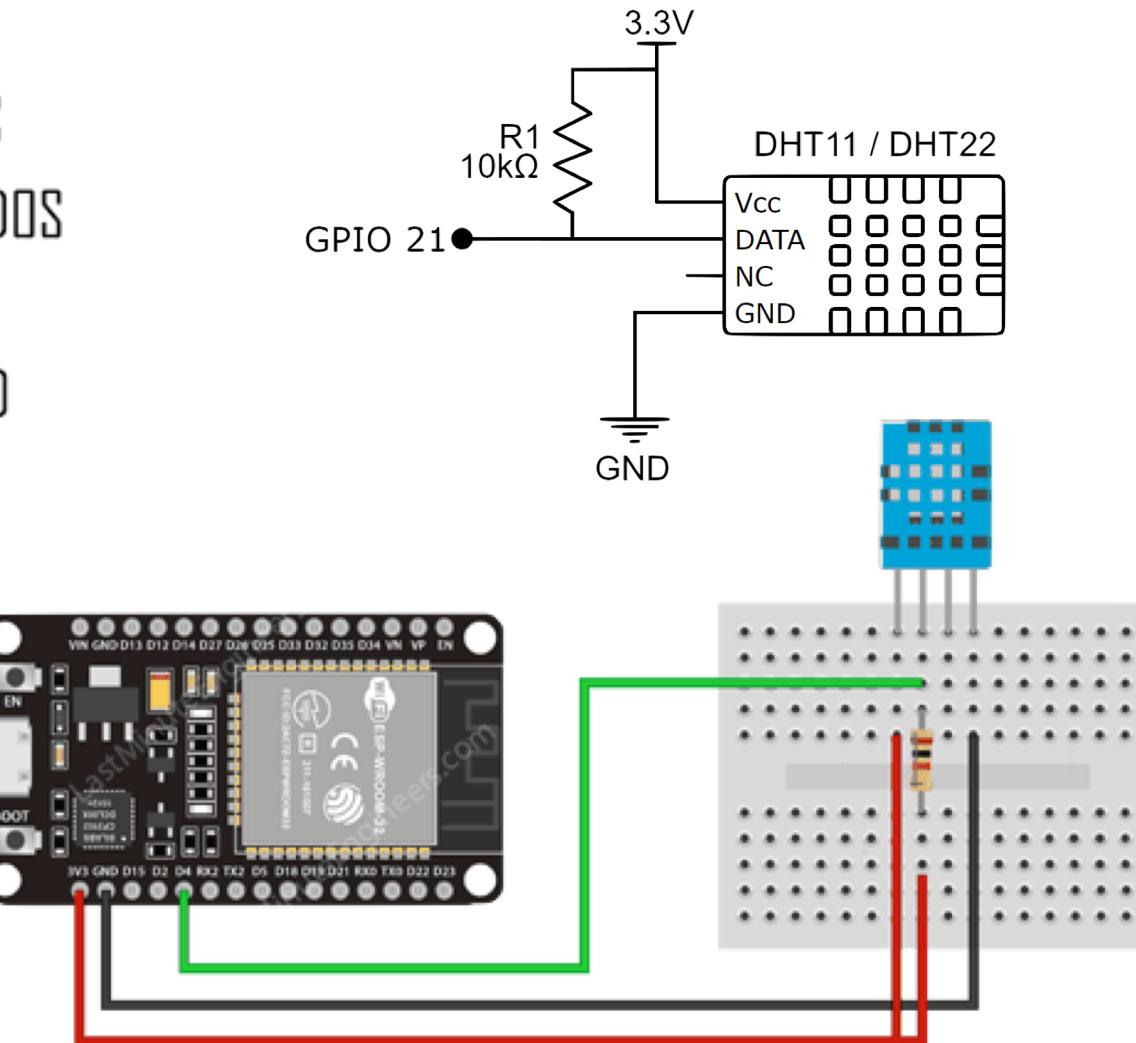
Sensor DHT11

- Muito baixo custo
- Tensão de alimentação de 3V a 5V
- 2.5mA de corrente máxima durante a conversão
- Bom para medir umidade entre 20% e 80%, com 5% de precisão
- Bom para medir temperaturas entre 0 e 50°C, com ±2°C de precisão
- Taxa de amostragem de até 1Hz (1 leitura por segundo)
- Dimensões: 15.5mm x 12mm x 5.5mm
- 4 pinos com 0.1" de espaçamento entre eles

Sensor DHT22

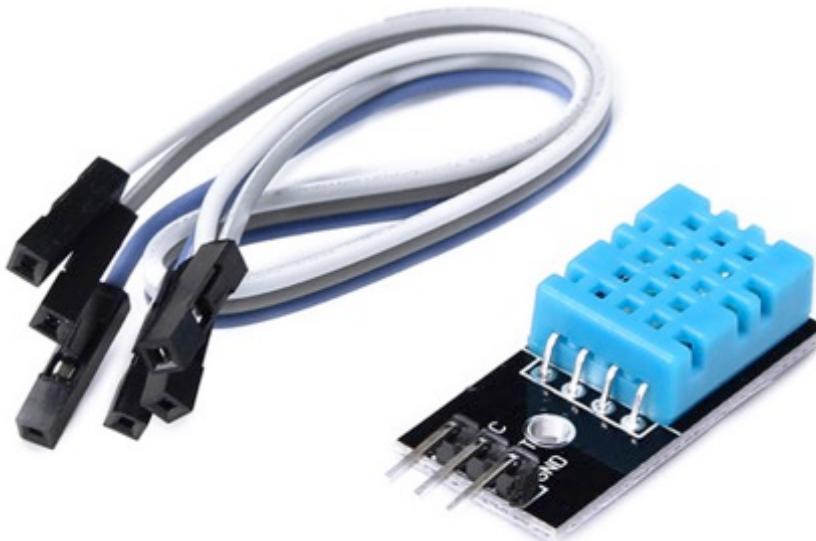
- Baixo custo
- Tensão de alimentação de 3V a 5V
- 2.5mA de corrente máxima durante a conversão
- Bom para medir umidade entre 0% e 100%, com 2% a 5% de precisão
- Bom para medir temperaturas entre -40 e 125°C, com ±0,5°C de precisão
- Taxa de amostragem de até 0,5Hz (2 leituras por segundo)
- Dimensões: 15.1mm x 25mm x 7.7mm
- 4 pinos com 0.1" de espaçamento entre eles

Sensor DHT11 e DHT22



Módulo DHT11

:: Características



Características:

- Voltagem de alimentação: 3,3 a 5 V
- Faixa de temperatura: 0 a 50 °C erro de ± 2 °C
- Umidade: 20 a 90% RH $\pm 5\%$ RH erro
- Interface: Digital

Instalação de bibliotecas

- Vá em *Sketch > Incluir Biblioteca > Gerenciar Bibliotecas* e pesquise por **Universal Telegram Bot** e instale a biblioteca
- Vá em *Sketch > Incluir Biblioteca > Gerenciar Bibliotecas* e pesquise por **ArduinoJson** e instale a biblioteca

Código ESP32

Código

```
#include <WiFi.h>
#include <WiFiClientSecure.h>
#include <UniversalTelegramBot.h>
#include "DHT.h"

#define WIFI_SSID "ICOMP_1A"
#define WIFI_PASSWORD "1comp@14#"

#define BOTtoken "5497495403:AAFxvkm3CdYx2ZnSj77mJbeIqHZ4y5FqctU"
```

Código

```
#define LED_PIN 22
#define DHT_PIN 23
#define DHTTYPE DHT11

#define BOT_SCAN 1000
long lastTimeScan;
bool ledStatus;

WiFiClientSecure client;
UniversalTelegramBot bot(BOTtoken, client);
DHT dht(DHT_PIN, DHTTYPE);
```

Código

```
void handleNewMessages(int numNewMessages) {  
    float humidity, temperature;  
    String message, chat_id, text, from_name;  
  
    for (int i=0; i<numNewMessages; i++) {  
        chat_id = String(bot.messages[i].chat_id);  
        from_name = bot.messages[i].from_name;  
        text = bot.messages[i].text;  
        Serial.println(text);  
  
        if (text == "/ledon") {  
            digitalWrite(LED_PIN, HIGH);  
            ledStatus = true;  
            bot.sendMessage(chat_id, "Led está ligado", "");  
            Serial.println("Led esta ligado");  
        }  
    }  
}
```

Código

```
if (text == "/ledoff") {  
    ledStatus = false;  
    digitalWrite(LED_PIN, LOW);  
    bot.sendMessage(chat_id, "Led está desligado", "");  
    Serial.println("Led esta desligado");  
}  
  
if( text == "/env") {  
    humidity = dht.readHumidity();  
    temperature = dht.readTemperature();  
    message = "A temperatura é de " + String(temperature, 2) + "  
graus celsius.\n";  
    message += "A umidade relativa do ar é de " + String(humidity,  
2)+ "%.\n";  
    bot.sendMessage(chat_id, message, "Markdown");  
    Serial.println(message);  
}
```

Código

```
if (text == "/status") {  
    String message = "Led está ";  
    if(ledStatus){  
        message += "ligado";  
    }else{  
        message += "desligado";  
    }  
    message += ". \n";  
    humidity = dht.readHumidity();  
    temperature = dht.readTemperature();  
    message += "A temperatura é de " + String(temperature, 2) + "  
graus celsius.\n";  
    message += "A umidade relativa do ar é de " + String(humidity,  
2)+ "%.\n";  
    bot.sendMessage(chat_id, message, "");  
    Serial.println(message);  
}
```

Código

```
if (text == "/start") {  
    String welcome = from_name + ", bem vindo ao Bot do  
        DevTitans do Barreto.\n";  
    welcome += "Para interagir, use um dos comandos a  
        seguir.\n";  
    welcome += "/ledon : para ligar o Led \n";  
    welcome += "/ledoff : para desligar o Led \n";  
    welcome += "/env : saber a temperatura e umidade do  
        ambiente \n";  
    welcome += "/status : para saber o status dos sensores  
        e atuadores \n";  
    bot.sendMessage(chat_id, welcome, "");  
}  
}
```

Código

```
void setupWifi(){  
    Serial.print("Connecting Wifi: ");  
    Serial.println(WIFI_SSID);  
    client.setCACert(TELEGRAM_CERTIFICATE_ROOT);  
    WiFi.begin(WIFI_SSID, WIFI_PASSWORD);  
  
    while (WiFi.status() != WL_CONNECTED) {  
        Serial.print(".");  
        delay(500);  
    }  
  
    Serial.println("");  
    Serial.println("WiFi connected");  
}
```

Código

```
void setupPins(){  
    pinMode(LED_PIN, OUTPUT);  
    pinMode(DHT_PIN, INPUT_PULLUP);  
    digitalWrite(LED_PIN, LOW);  
    ledStatus = false;  
    dht.begin();  
}
```

Código

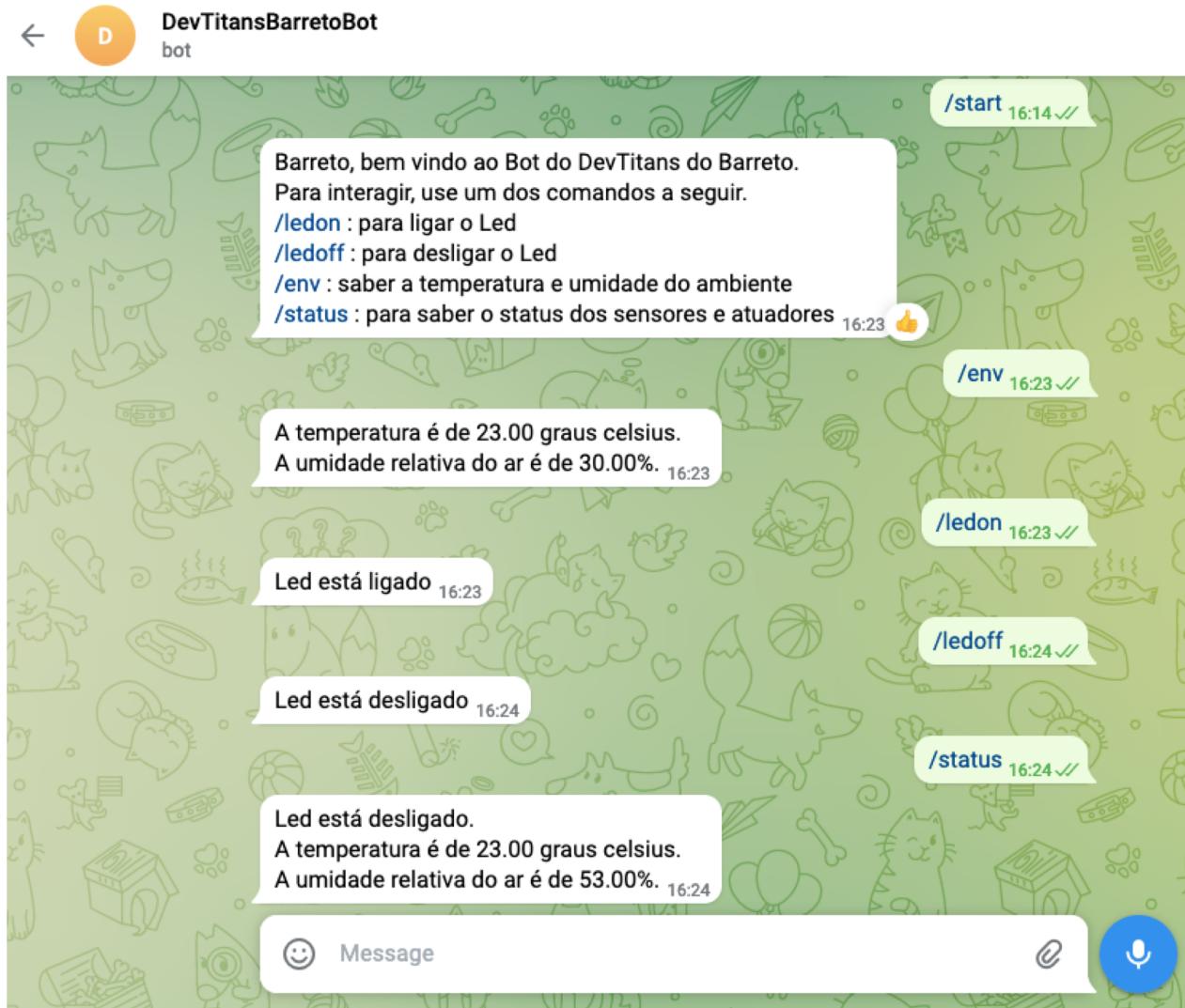
```
void setup() {  
    Serial.begin(115200);  
    WiFi.mode(WIFI_STA);  
    setupWifi();  
    setupPins();  
    lastTimeScan = millis();  
}
```

Código

```
void loop() {  
    if (millis() > lastTimeScan + BOT_SCAN) {  
        int numNewMessages = bot.getUpdates(bot.last_message_received  
                                         + 1);  
        while(numNewMessages) {  
            handleNewMessages(numNewMessages);  
            numNewMessages = bot.getUpdates(bot.last_message_received  
                                         + 1);  
        }  
        lastTimeScan = millis();  
    }  
}
```

Execução

Execução



Execução

A screenshot of a terminal window titled "/dev/cu.usbserial-0001". The window contains the following text:

```
/start
/env
A temperatura é de 23.00 graus celsius.
A umidade relativa do ar é de 30.00%.
/ledon
Led esta ligado
/ledoff
Led esta desligado
/status
Led está desligado.
A temperatura é de 23.00 graus celsius.
A umidade relativa do ar é de 53.00%.
```

The terminal has several status indicators at the bottom:

- Auto-rolagem Show timestamp
- Nova-linha
- 115200 velocidade
- Deleta a saída



Um pouco de segurança

Mais seguro

- Qualquer pessoa que conheça o nome de usuário do seu bot pode interagir com ele
- Para garantir que ignoramos mensagens que não são da nossa conta do Telegram, você pode obter sua ID de usuário do Telegram
- Quando seu bot recebe uma mensagem, o ESP pode verificar se o ID do remetente corresponde ao seu ID de usuário e lidar com a mensagem ou ignorá-la
- É possível também criar grupos de usuários

Mais seguro

- Na sua conta do Telegram, procure por “IDBot” ou abra este link t.me/myidbot em seu smartphone
- Inicie uma conversa com esse bot e digite /getid
- Você receberá uma resposta de volta com seu ID de usuário, que deve ser salvo para uso posterior

Mais seguro

The screenshot shows a Firefox browser window with the following details:

- Toolbar:** Firefox, Arquivo, Editar, Exibir, Histórico, Favoritos, Ferramentas, Janela, Ajuda.
- Address Bar:** https://web.telegram.org/k/#@myidbot
- Content Area:**
 - IDBot** bot profile: Your own ID is: [REDACTED] (7:12)
 - Recent Chats:**
 - ID Telegram Bot /getid (7:11)
 - DevTitansBarretoBot /options (7:04)
 - DevTITANS 01 Lucas left the group (03:36)
 - Mateus Preste UFAM Tenho sim. Levo amanhã... (7:12)
 - BotFather 🤖 Done! Congratulations on your new bot. Yo... (7:12)
 - Edrem TV Amazonas Edrem TV Amazonas joined Telegram (7:12)
 - Leonardo CESAR Leonardo CESAR joined Telegram (7:12)
 - Telegram 📡 Para não cair em fake news sobre...
 - Info Box:** What can this bot do?

Hi!
I'm your loyal Telegram ID Assistant, and can supply you with your Telegram user/group/channel ID.
Use `/getid` command to get your Telegram ID
Or alternatively, use `/getgroupid` to get the the ID of a group.

You can also forward a message me, and if the privacy settings of the sender allow it, I will recover the ID of the sender for you.
 - Message History:**
 - Today: Barreto /start 07:12
 - I'm eager to help you! Please send a valid command. 07:12
 - Barreto /getid 07:12
 - Your own ID is: [REDACTED] 07:12
 - Bottom Bar:** Message input field with icons for attachments, smiley, and microphone.

Mais seguro

```
#include <WiFi.h>
#include <WiFiClientSecure.h>
#include <UniversalTelegramBot.h>
#include "DHT.h"

#define WIFI_SSID "ICOMP_1A"
#define WIFI_PASSWORD "1comp@14#"

#define BOTtoken "5497495403:AAFxvkm3CdYx2ZnSj77mJbeIqHZ4y5FqctU"

#define CHAT_ID "XXXXXXXXXX"
```

Mais seguro

```
void handleNewMessages(int numNewMessages) {  
    float humidity, temperature;  
    String message, chat_id, text, from_name;  
  
    for (int i=0; i<numNewMessages; i++) {  
        chat_id = String(bot.messages[i].chat_id);  
        if (chat_id != CHAT_ID){  
            bot.sendMessage(chat_id, "Unauthorized user", "");  
            continue;  
        }  
        from_name = bot.messages[i].from_name;  
        text = bot.messages[i].text;  
        Serial.println(text);
```