

Manual Criado por André Jacob com apoio de Álvaro Ramos 05/07/2020.

Referência: Github do desenvolvedor - <https://bit.ly/2FVYVVK>

Canal no YouTube: <https://www.youtube.com/c/JacobTecnologia>

Instagram: <https://www.instagram.com/andrejacob13/>

Facebook: <https://www.facebook.com/andrejacobh>

Google Home Grátis com Esp8266 NodMCu

Nesse manual você vai aprender a Acender as Lâmpadas da sua casa por comando de voz, estando em qualquer lugar do mundo!

O manual está dividido em 4 passos:

Passo 1 – Criação das chaves.

Passo 2 – O código.

Passo 3 – Criação da sua Nuvem.

Passo 4 – Baixando e configurando os aplicativos Google.

PASSO 1 (criação das chaves)

- 1) Crie uma conta no site: sinric.com Clique em Register pra criar uma nova conta:

The screenshot shows the Sinric.com homepage. At the top is the Sinric logo and the text 'Welcome to Sinric.com'. Below this is a description: 'Simple, elegant way to link your development boards like RaspberryPi, ESP8226, ESP32 or Arduino with Amazon Alexa or Google Home for FREE!'. A message follows: 'After months of testing we're proud to introduce our Sinric V2 !. Please migrate your existing code to the newer version to avoid disruption. For more details visit [here](#)'. There are two sections: 'How to use: Alexa' and 'How to use: Google Home?'. The Alexa section has a list of steps: 1. Register for an account if you do not have one (Use Chrome, FireFox), 2. Login and create a smart home device, 3. Copy your API Key, 4. Connect. Example [Switch](#), 5. Install Amazon Alexa smart home skill (US, UK, Germany, Australia, India, Italy, Spanish (ES/MX), Canada, Japan) [here](#), 6. Buy me a beer! paypal.me/arunat. The Google Home section has a link 'Click here!' and text 'Or support me by buying a [IRDEVKIT](#) to turn on or off your TV or air conditioner via Amazon Alexa'. On the right, there is a 'Please Login' form with fields for 'Enter your email *' (with a red error message 'You must enter a value') and 'Enter your password' (with a toggle icon). Below the fields are 'Login' and 'Register' buttons. A large green arrow points to the 'Register' button. At the bottom, there is text 'Got suggestions or an issue? Please create an issue in our GitHub repository', 'Trusted by 17820 users', 'For documentation', and 'Download the app'.

- 2) Digite: Primeiro Nome; Sobrenome; email e senha e clique em submit.

Register

First Name *
 Andre

Last Name *
 Jacob de Oliveira

Email *
 andrejacob856@gmail.com

Password *

Submit

3) Apareça essa mensagem:

Registration successful

4) Agora entre na sua conta criada com a senha e o email que você registrou.

5) Clique em Add para adicionar um dispositivo:

← → ↻ sinric.com

Apps

Hi Andre!

Your API Key

Keep your API key secure

c06f35b2-664a-4d1b-a18a-f4928a1dc64a

COPY

Smart Home Device

You can link with Amazon Alexa or Google Home

Click on Add button to create a smart home device

Add

Google Home

How to integrate with Google Home [here](#)

Client ID : Click 'Generate' button

Client Secret : Click 'Generate' button

Google HomeGraph API Key (Optional)

Generate Keys Save API Key

6) Vai aparecer essa tela: Nela digite o nome da aplicação e o tipo de dispositivo que é, nessa caso uma lâmpada e em device type coloque Light

New Device

Friendly Name (Alexa invocation name) *

Iluminação

Description

Device Type

Light

Save

- 7) Clique em Save. (repita a operação para cadastrar mais dispositivos). No exemplo fiz com 2.

- 8) Clique em Generate keys e em Test e será gerado as chaves e exibido o sucesso na execução da atividade. Essas chaves utilizaremos no decorrer do manual.

Hi Andre!

Success!

Your API Key

Keep your API key secure

c06f35b2-664a-4d1b-a18a-f4928a1dc64a

COPY

Smart Home Device

You can link with Amazon Alexa or Google Home

Click on Add button to create a smart home device

Add

Google Home

How to integrate with Google Home [here](#)

Client ID : Skop4faRL

Client Secret : 9f50f8a9-ef03-474a-9af1-3da8d86beccf

Google HomeGraph API Key (Optional)

Generate Keys

Save API Key

Your devices

Iluminação (Light)

Device ID : 5eff939fad7a48327f302926

Test

Ventilador (Switch)

Device ID : 5eff93dcad7a48327f302934

Test

Passo 2 (O código)

- 1) Insira esse código na IDE do Arduino

```
#include <Arduino.h>
#include <ESP8266WiFi.h>
#include <ESP8266WiFiMulti.h>
#include <WebSocketsClient.h> // https://github.com/kakopappa/sinric/wiki/How-to-add-dependency-libraries
#include <ArduinoJson.h> // https://github.com/kakopappa/sinric/wiki/How-to-add-dependency-libraries
#include <StreamString.h>
```

```
ESP8266WiFiMulti WiFiMulti;
WebSocketsClient webSocket;
WiFiClient client;
```

```
#define MyApiKey "Insira a sua chave API"
#define MySSID "Insira o nome da sua rede WiFi"
#define MyWifiPassword "Insira a senha do seu WIFI"
#define HEARTBEAT_INTERVAL 300000 // 5 Minutes
#define D1 5
#define D2 4
```

```
uint64_t heartbeatTimestamp = 0;
bool isConnected = false;
```

```
void turnOn(String deviceId) {
  if (deviceId == "Insira o ID do seu primeiro dispositivo") // Device ID of first device
  {
    Serial.print("Turn on device id: ");
    Serial.println(deviceId);

    digitalWrite(D1, HIGH);
  }
  else if (deviceId == "Insira o ID do seu segundo Dispositivo") // Device ID of second device
  {
    Serial.print("Turn on device id: ");
    Serial.println(deviceId);
    digitalWrite(D2, HIGH);
  }
  else {
    Serial.print("Turn on for unknown device id: ");
    Serial.println(deviceId);
  }
}
```

```
void turnOff(String deviceId) {
  if (deviceId == "Insira o ID do Primeiro Dispositivo") // Device ID of first device
  {
    Serial.print("Turn off Device ID: ");
    Serial.println(deviceId);
    digitalWrite(D1, LOW);
  }
  else if (deviceId == "Insira o ID do segundo Dispositivo") // Device ID of second device
  {
    Serial.print("Turn off Device ID: ");
```

```

    Serial.println(deviceId);
    digitalWrite(D2, LOW);
}
else {
    Serial.print("Turn off for unknown device id: ");
    Serial.println(deviceId);
}
}

void websocketEvent(WStype_t type, uint8_t * payload, size_t length) {
    switch (type) {
        case WStype_DISCONNECTED:
            isConnected = false;
            Serial.printf("[WSc] Webservice disconnected from sinric.com!\n");
            break;
        case WStype_CONNECTED: {
            isConnected = true;
            Serial.printf("[WSc] Service connected to sinric.com at url: %s\n", payload);
            Serial.printf("Waiting for commands from sinric.com ...\n");
        }
        break;
        case WStype_TEXT: {
            Serial.printf("[WSc] get text: %s\n", payload);
            // Example payloads

            // For Switch types
            // {"deviceId":"xxx","action":"action.devices.commands.OnOff","value":{"on":true}} //
https://developers.google.com/actions/smarthome/traits/onoff
            // {"deviceId":"xxx","action":"action.devices.commands.OnOff","value":{"on":false}}

#ifdef ARDUINOJSON_VERSION_MAJOR == 5
            DynamicJsonBuffer jsonBuffer;
            JsonObject& json = jsonBuffer.parseObject((char*)payload);
#elseif
#ifdef ARDUINOJSON_VERSION_MAJOR == 6
            DynamicJsonDocument json(1024);
            deserializeJson(json, (char*) payload);
#elseif
            String deviceId = json ["deviceId"];
            String action = json ["action"];

            if (action == "action.devices.commands.OnOff") { // Switch
                String value = json ["value"]["on"];
                Serial.println(value);

                if (value == "true") {
                    turnOn(deviceId);
                } else {
                    turnOff(deviceId);
                }
            }
            else if (action == "test") {
                Serial.println("[WSc] received test command from sinric.com");
            }
        }
    }
}

```

```

        break;
    case WStype_BIN:
        Serial.printf("[WSc] get binary length: %u\n", length);
        break;
    default: break;
}
}

void setup() {
    Serial.begin(115200);
    pinMode(D1, OUTPUT);
    pinMode(D2, OUTPUT);

    WiFiMulti.addAP(MySSID, MyWifiPassword);
    Serial.println();
    Serial.print("Connecting to Wifi: ");
    Serial.println(MySSID);

    // Waiting for Wifi connect
    while (WiFiMulti.run() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }
    if (WiFiMulti.run() == WL_CONNECTED) {
        Serial.println("");
        Serial.print("WiFi connected. ");
        Serial.print("IP address: ");
        Serial.println(WiFi.localIP());
    }

    // server address, port and URL
    websocket.begin("iot.sinric.com", 80, "/"); // "iot.sinric.com", 80

    // event handler
    websocket.onEvent(webSocketEvent);
    websocket.setAuthorization("apikey", MyApiKey);

    // try again every 5000ms if connection has failed
    websocket.setReconnectInterval(5000); // If you see 'class WebSocketsClient' has no member named
    'setReconnectInterval' error update arduinoWebSockets
}

void loop() {
    websocket.loop();

    if (isConnected) {
        uint64_t now = millis();

        // Send heartbeat in order to avoid disconnections during ISP resetting IPs over night. Thanks
        @MacSass
        if ((now - heartbeatTimestamp) > HEARTBEAT_INTERVAL) {
            heartbeatTimestamp = now;
            websocket.sendTXT("H");
        }
    }
}

```

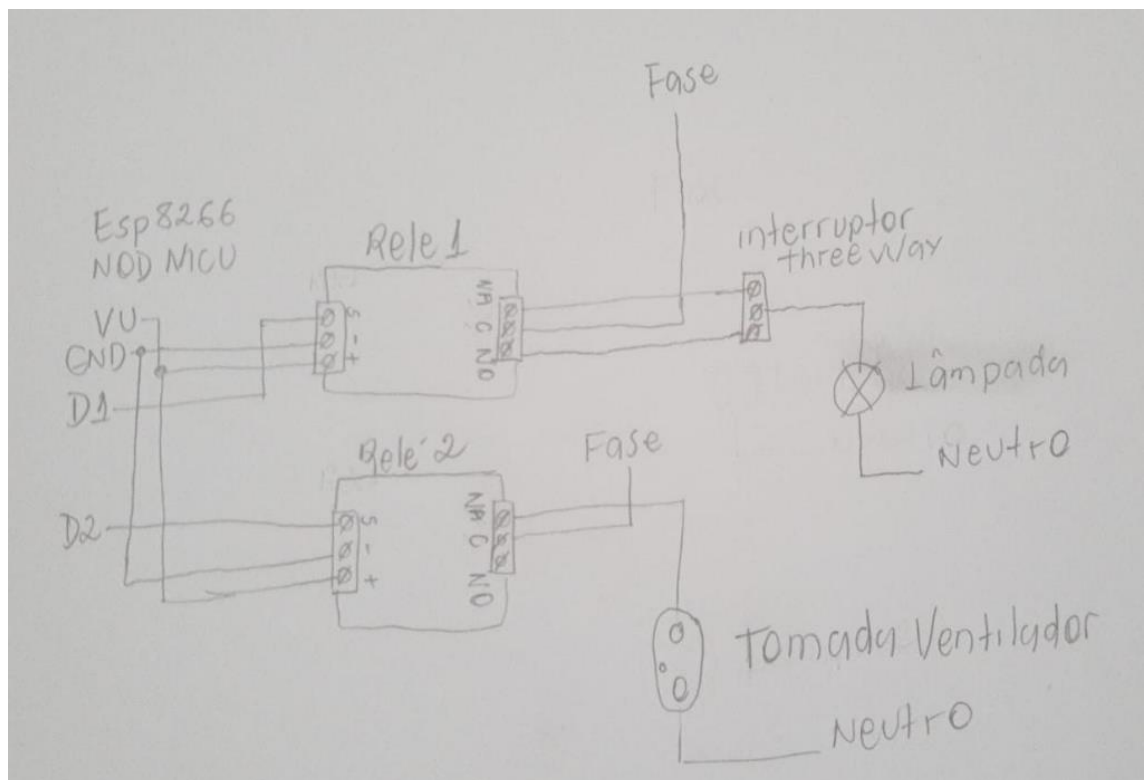
```
}
```

```
//fim do código
```

No código você deve mudar as seguintes informações antes de carregar:

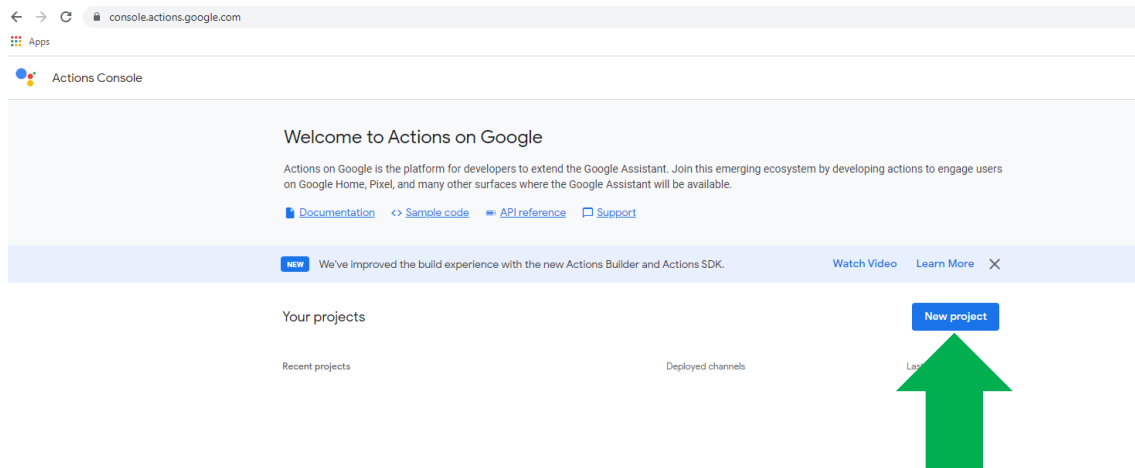
- 2) **Adicionar as bibliotecas:** Os links de cada biblioteca estão dentro do código.
- 3) **Inserir a API Key** - gerada no sinric.com
- 4) **Inserir o nome da sua rede WIFI**
- 5) **Inserir a senha da sua rede Wifi**
- 6) **Substituir o ID do dispositivo 1** – gerada no sinric.com – tem ele duas vezes no programa.
Substituir o ID do dispositivo 2 – gerada no sinric.com – tem ele duas vezes no programa.

- 7) O Sketch que montei no exemplo foi esse:



PASSO 3 (A criação da sua Nuvem)

1) Vai em <https://console.actions.google.com/>



2) Clique em New Project e Crie com um nome a sua escolha.

New Project


Project Name

Meu Manual

Choose a language for your action (you can change later)English

Choose your country or regionUnited States

CancelCreate project



3) Clique em Smart Home e depois em Start Building

Meu Manual

Get started


Start building

What kind of Action do you want to build?

Select the category that best fits the type of experience you want to build for the Google Assistant.


Smart Home

Let users control your smart home devices with the Google Assistant and the Google Home app




Game


Build anything from a trivia game to a fully immersive, multiplayer gaming experience



Custom

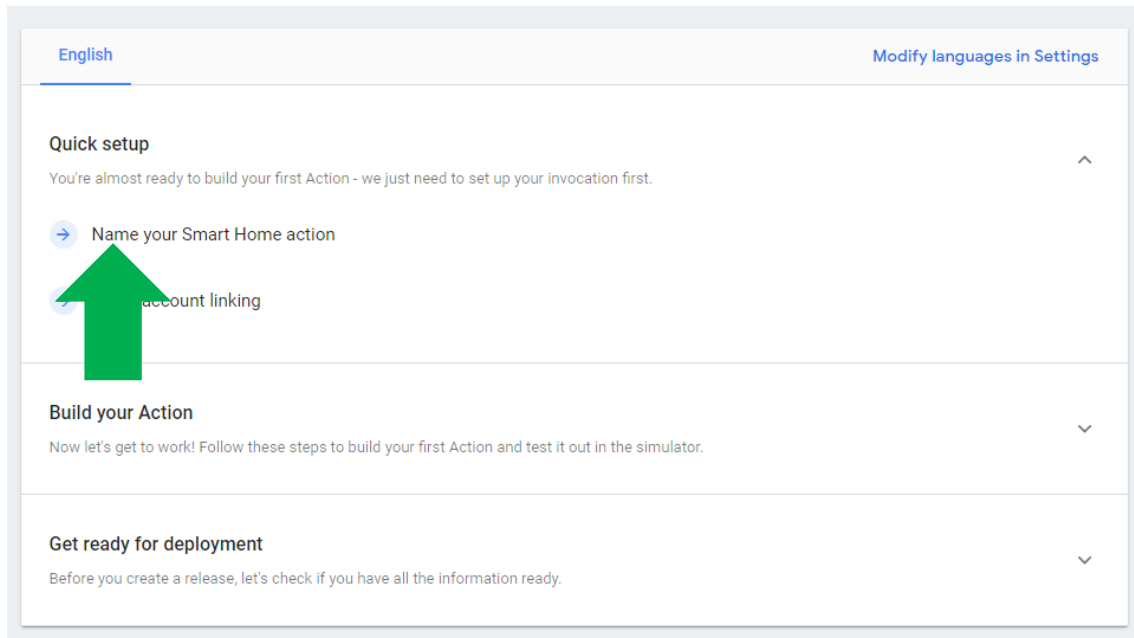
Don't see your category? Build a unique conversational experience for your users





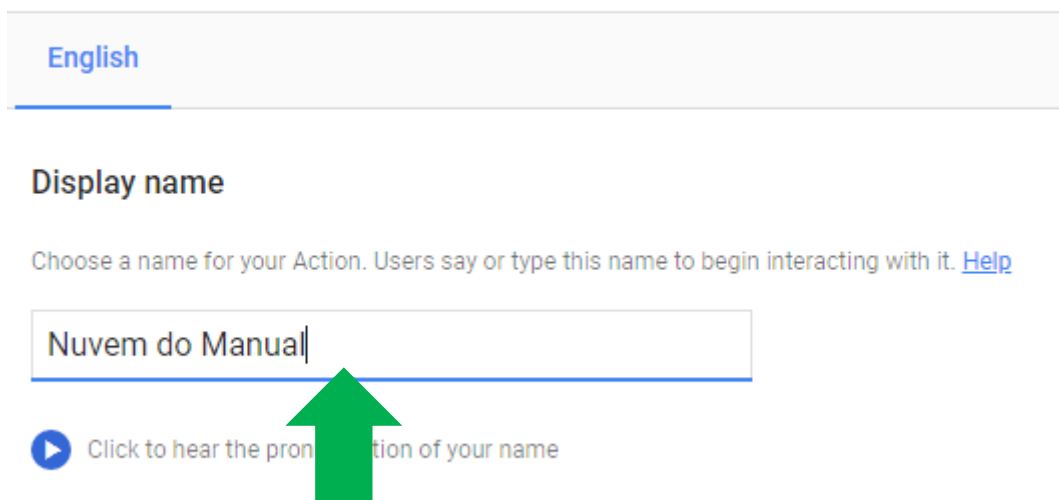
Need help? [Learn more](#) about building Actions for the Google Assistant

4) Agora em Quick setup, clique em Name your Smart Home action

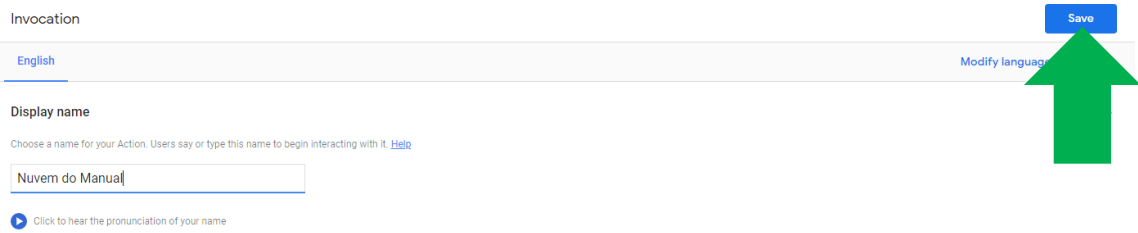


5) Crie um nome a sua escolha

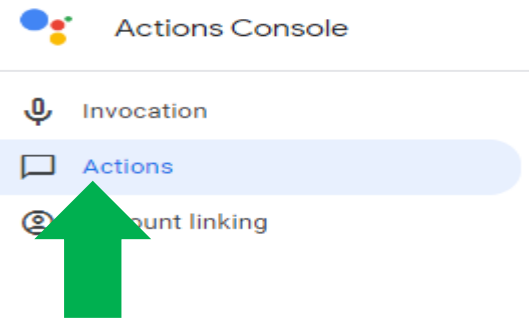
Invocation



6) Clique em Salvar no canto superior direito.



7) Clique em action do lado esquerdo da tela.



8) Cole esse link <https://sinric.com/api/v1/googlehome/event> em Fulfillment URL

Fulfillment URL

Enter the webhook used to process your smart home intents. [Learn more](#)

Configure local home SDK (optional)

Add local fulfillment to route your smart home intents. To configure the Local Home SDK, upload your JS files and add your scan configuration below. [Learn more](#)

Enter your testing URL for Chrome

Enter your testing URL for Node

Upload Javascript files

Latest SDK version

Upload JavaScript files

Add device scan configuration

+ New scan config

9) Clique em Salvar no canto superior esquerdo.

Fulfillment URL

Enter the webhook used to process your smart home intents. [Learn more](#)

Configure local home SDK (optional)

Add local fulfillment to route your smart home intents. To configure the Local Home SDK, upload your JS files and add your scan configuration below. [Learn more](#)

Enter your testing URL for Chrome

Enter your testing URL for Node

Upload Javascript files

Latest SDK version

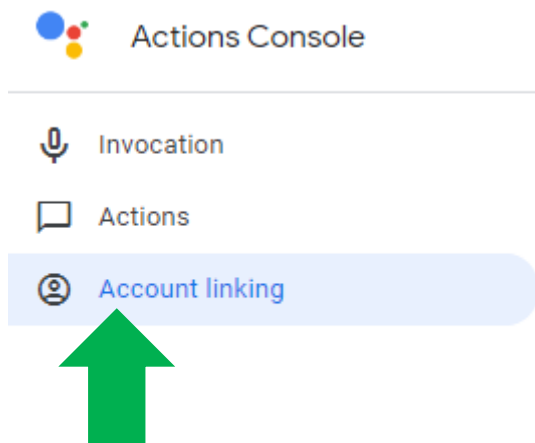
Upload JavaScript files

Add device scan configuration

+ New scan config

Save

10) Clique em Account Linking



11) Na próxima imagem:

Em Authorization URL, coloque esse link: <https://google.sinric.com/oauth>

Em Token URL, esse outro link: <https://google.sinric.com/token>

Account linking

1

OAuth Client Information


Client ID issued by your Actions to Google ⓘ

Sywew8AR8


Client secret ⓘ

.....

Authorization URL ⓘ

https://

Token URL ⓘ

https://

Back

Next

2

Configure your client (optional)

12) Clique em Salvar no canto inferior direito.

Account linking

1 OAuth Client Information

Client ID issued by your Actions to Google ⓘ

Sywew8AR8

Client secret ⓘ

.....

Authorization URL ⓘ

https://google.sinric.com/oauth


Token URL ⓘ

https://google.sinric.com/token

Back Next

✓ Configure your client (optional)

Cancel Save



13) Agora clique em teste no canto superior direito.

Account linking

OAuth Client Information

Client ID issued by your Actions to Google ⓘ

Sywew8AR8

Client secret ⓘ

.....

Authorization URL ⓘ


https://google.sinric.com/oauth

Token URL ⓘ

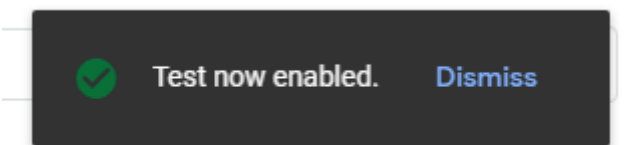
https://google.sinric.com/token

Configure your client (optional)

Test Save

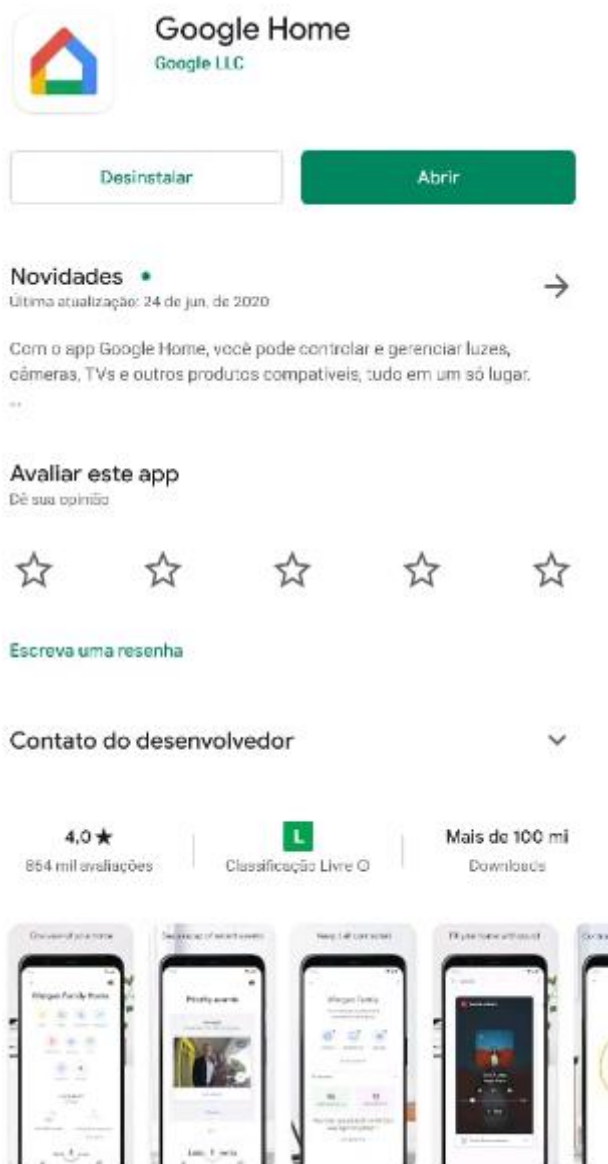


14) Caso esteja tudo certo vai aparecer uma mensagem assim:

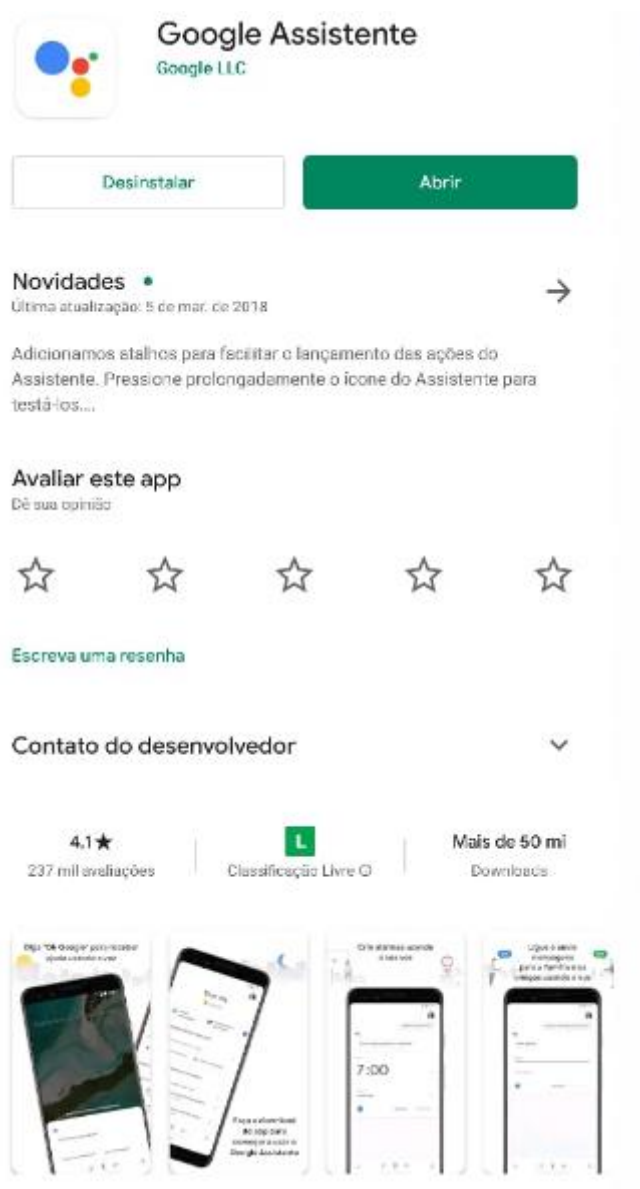


Passo 4 (Baixar os aplicativos no Smartphone e Configurar).

1) Vamos agora baixar o aplicativo Google Home no smartphone

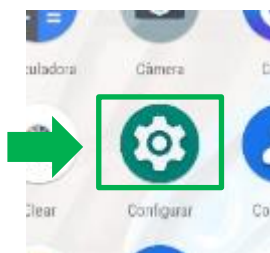


2) Vamos agora baixar o aplicativo Google Assistente no smartphone.



Vamos agora configurar o Google home:

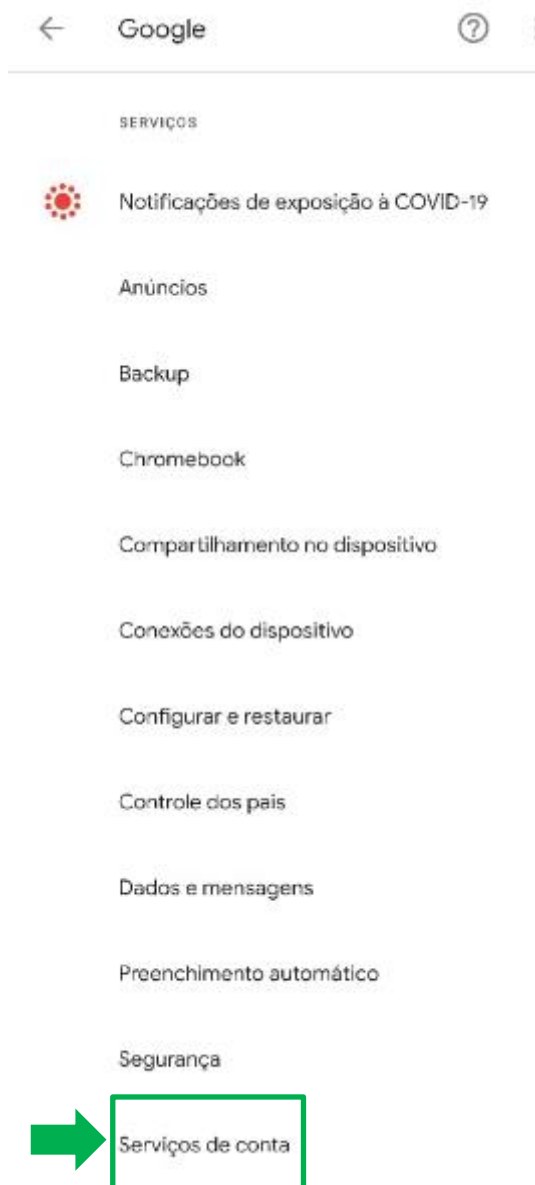
3) Abra configurações do seu smartphone.



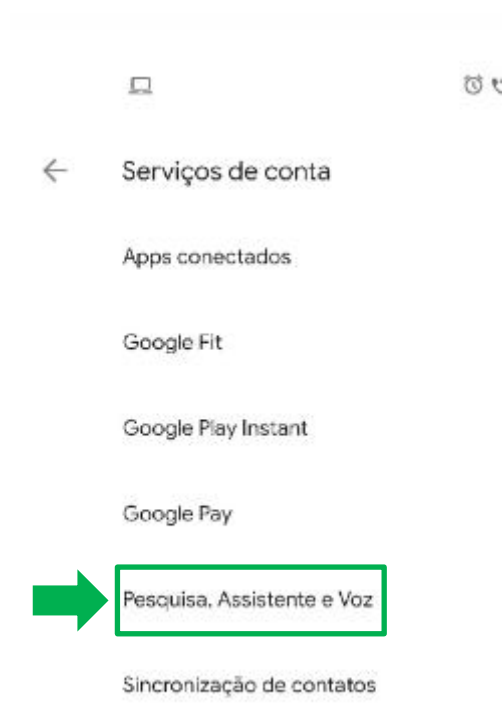
4) Vá em Google – Serviços e preferências dentro de configurações.



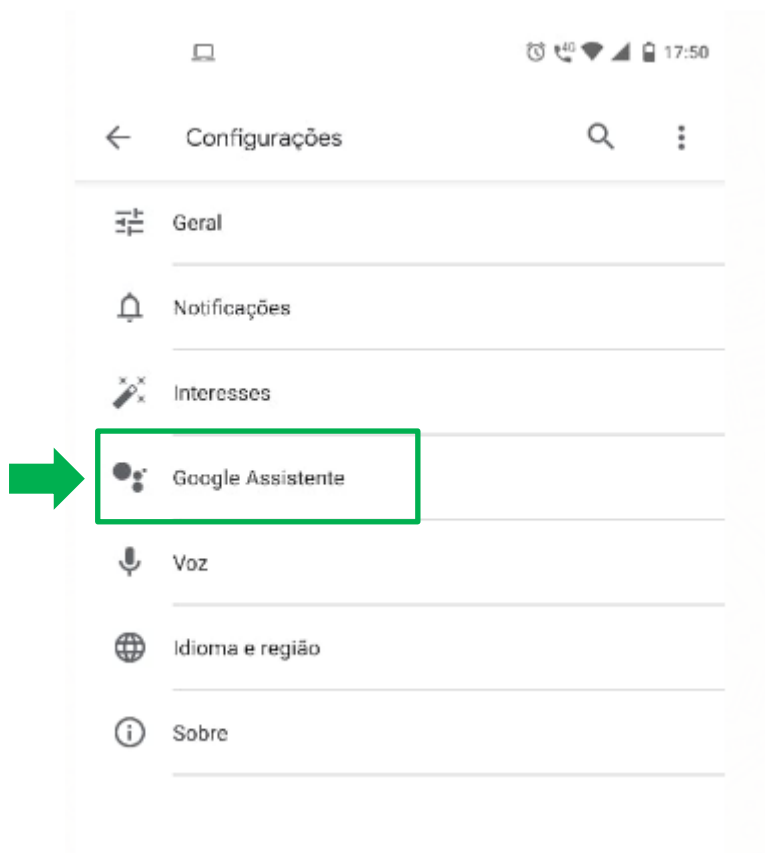
5) Vai em Serviços de Conta.



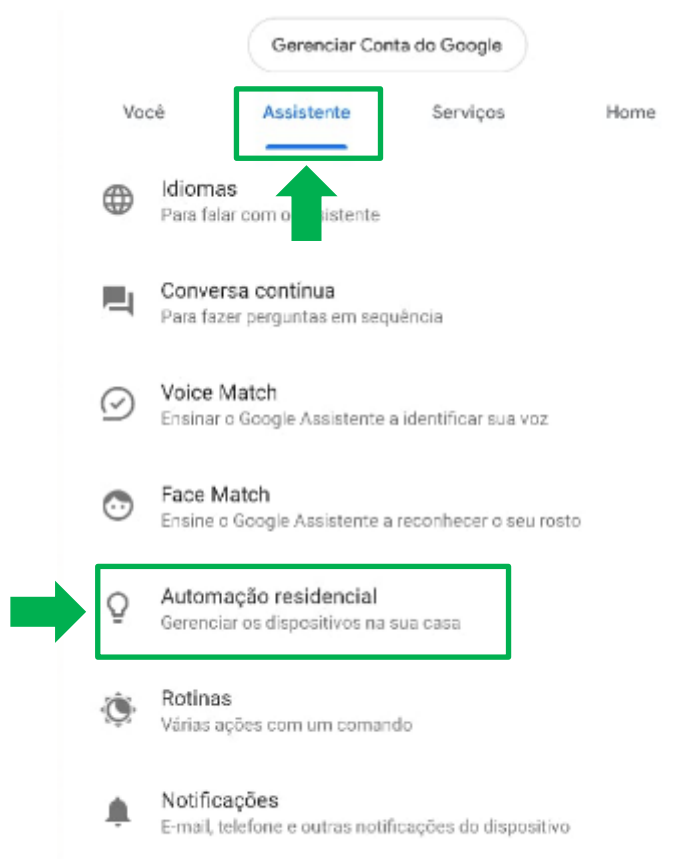
6) Vai Pesquisa Assistente e Voz.



7) Vai em Google Assistente.



8) Clica em Assistente e depois em Automação Residencial

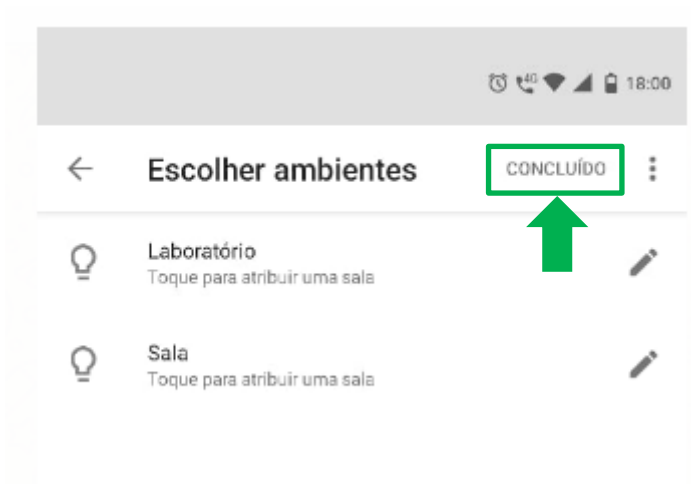


Dispositivos - Assistente

9) Clica nessa nuvem de contorno preto com fundo branco, com o nome que você criou no site consoleaction, ela vai vincular o que você já fez com o seu dispositivo. Primeiro em baixo como na seta e depois acima.



10) Clique agora em concluído e depois clique em OK.



11) Agora pode abrir o Google Home e teste comandos de acender e apagar.

Teste o comando de voz como ligar tudo, desligar tudo, acender lâmpada... de acordo com o seu ambiente.

Alguns detalhes do projeto podem ser diferentes no seu smartphone, mas é só dedicar um tempo aí que vai dar tudo certo. Espero que tenha ajudado, pois deu trabalho pra fazer! Valeu!