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.

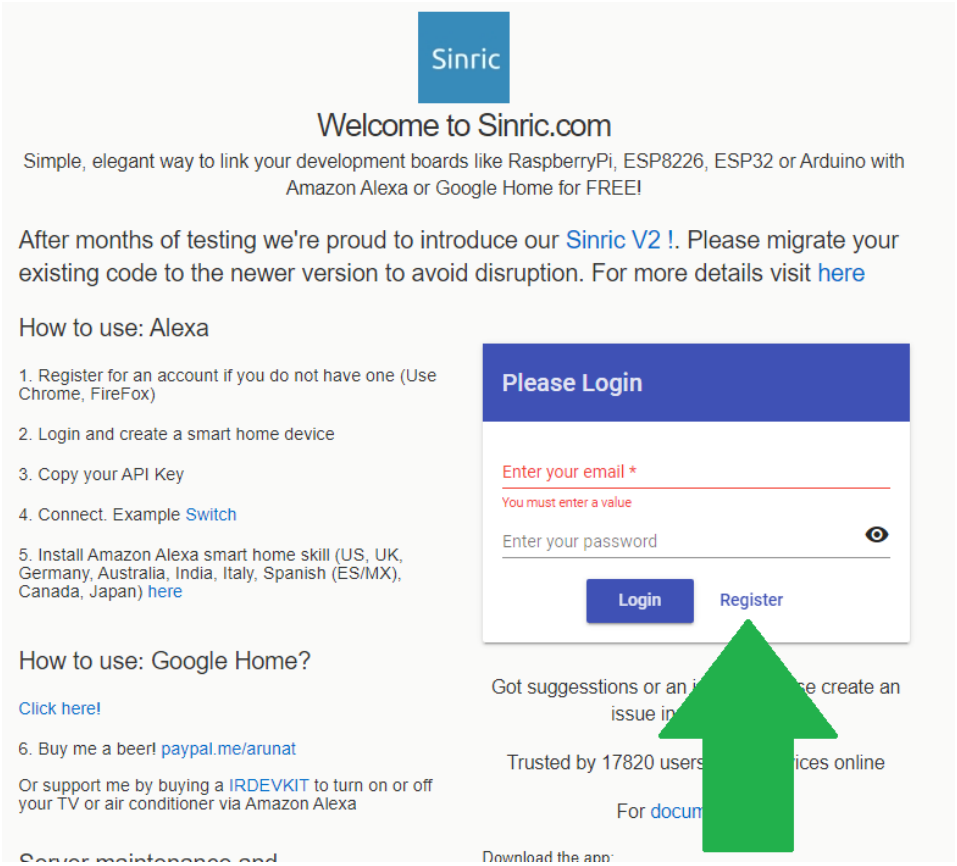
Passo 2 – O código.

Passo 3 – Criação da sua Nuvem.

Passo 4 – Baixando e configurando os aplicativos Google.

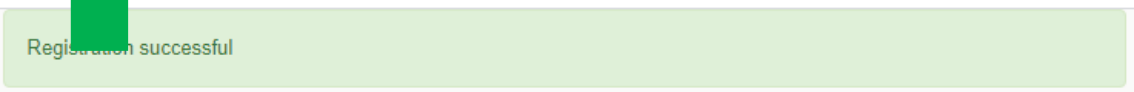
PASSO 1 (criação das chaves)

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

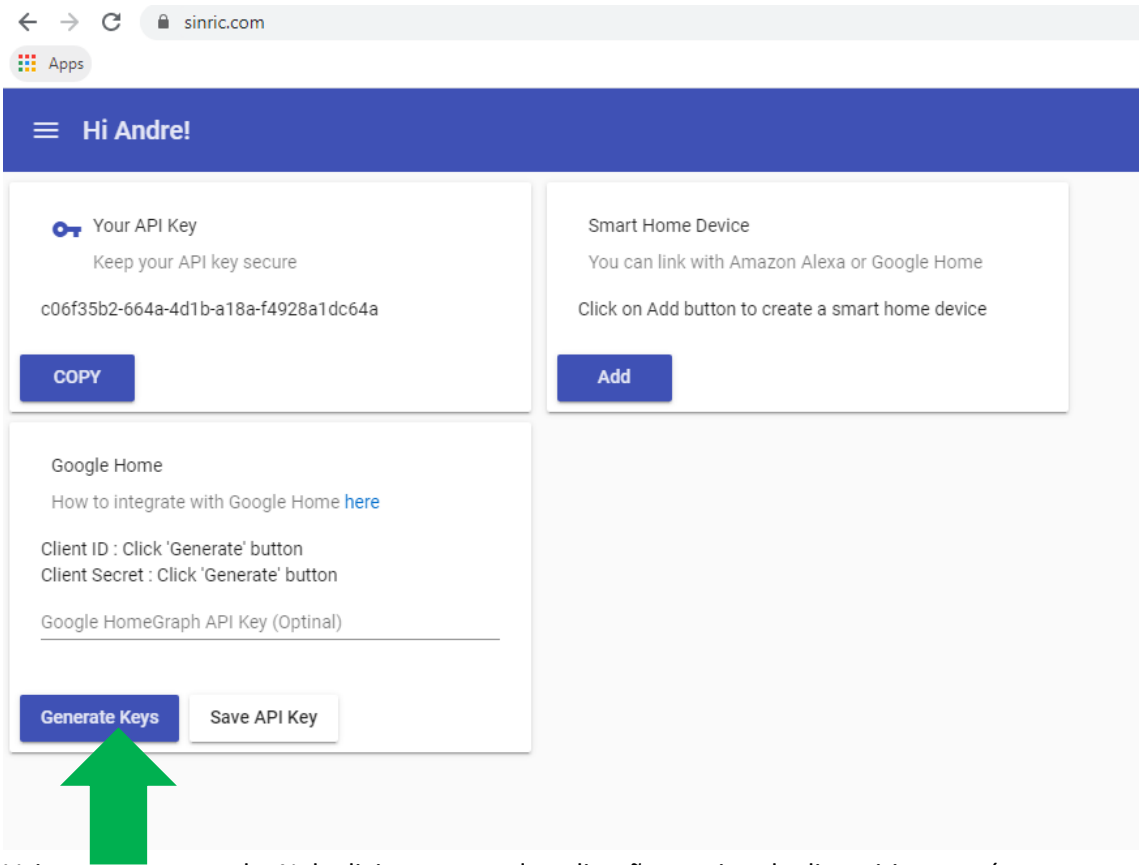


2) Digite: Primeiro Nome; Sobrenome; email e senha e clique em submit.

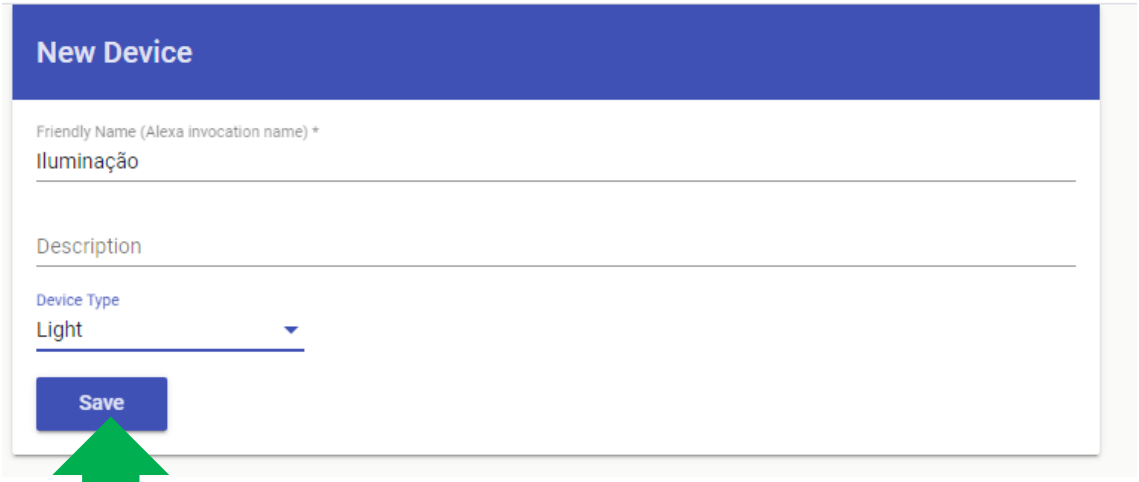
3) Apare essa mensagem:



4) Agora entre na sua conta criada com a senha e o email que você registrou.
5) Clique em Add para adicionar um dispositivo:

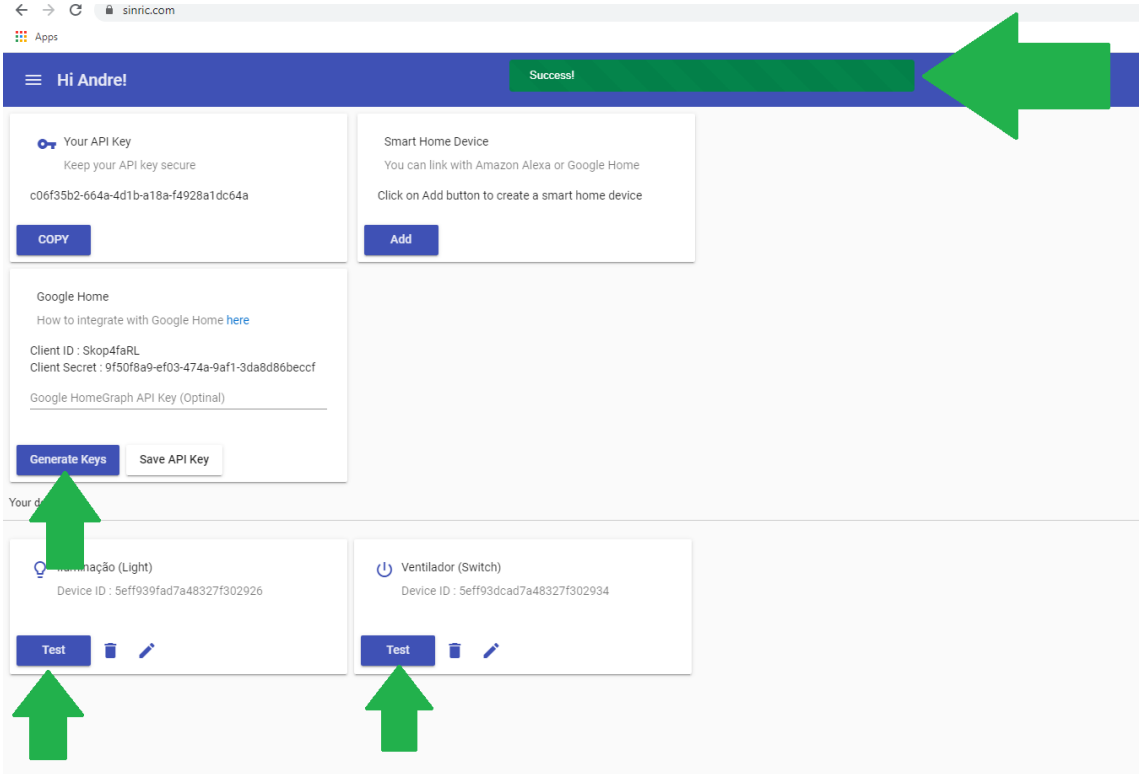


6) Vai aparecer essa tela: Nela digite o nome da aplicação e o tipo de dispositivo que é, nessa caso uma lâmpada e em device type coloque Light



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

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



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}}

#if ARDUINOJSON_VERSION_MAJOR == 5
       DynamicJsonBuffer jsonBuffer;
       JsonObject& json = jsonBuffer.parseObject((char*)payload);
#endif
#if ARDUINOJSON_VERSION_MAJOR == 6
       DynamicJsonDocument json(1024);
       deserializeJson(json, (char*) payload);
#endif
       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");
    }
  }
```

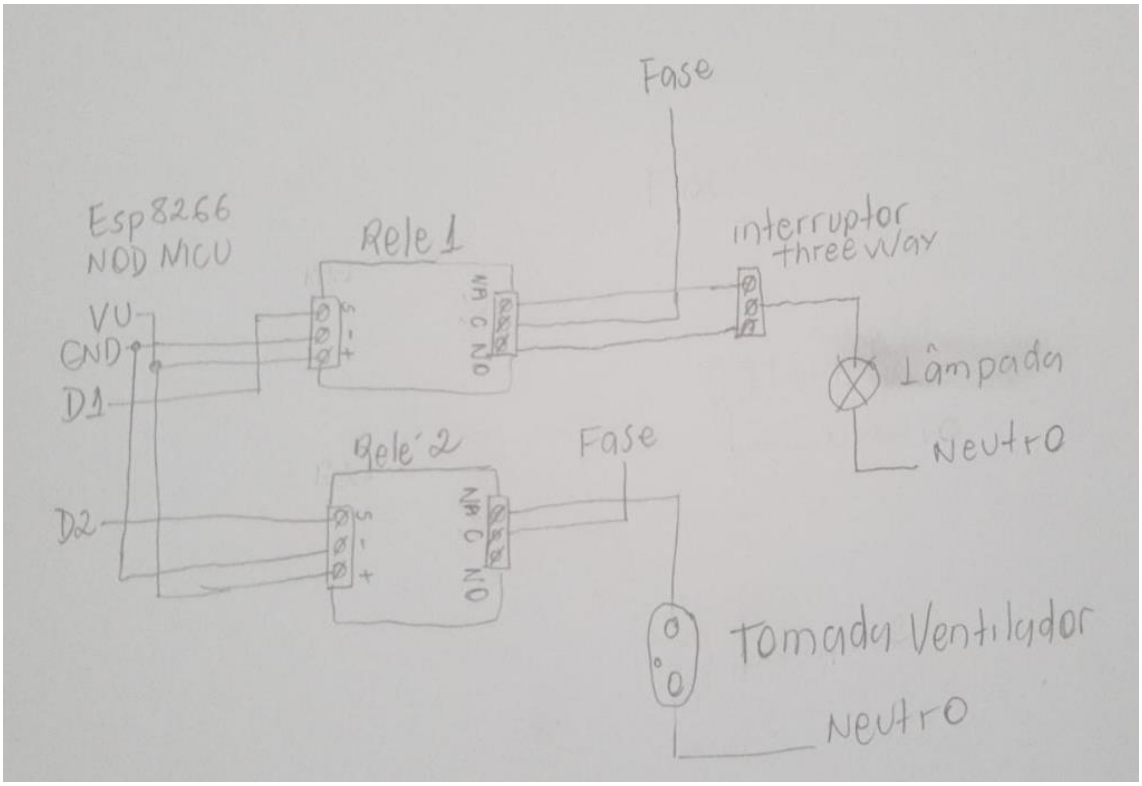
```
}

//fim do código
```

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

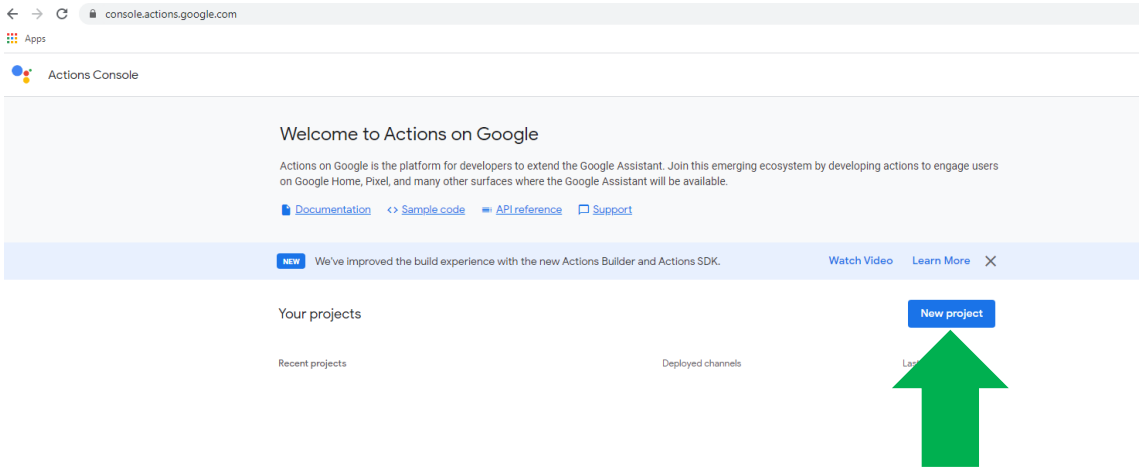
2) Adicionar as bibliotecas: Os links de cada biblioteca estão dentro do código.
3) Inserir a API Key - gerada no sinric.com
4) Inserir o nome da sua rede WIFI
5) Inserir a senha da sua rede Wifi
6) Substituir o ID do dispositivo 1 – gerada no sinric.com – tem ele duas vezes no programa.
   Substituir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

7) O Sketch que montei no exemplo foi esse:

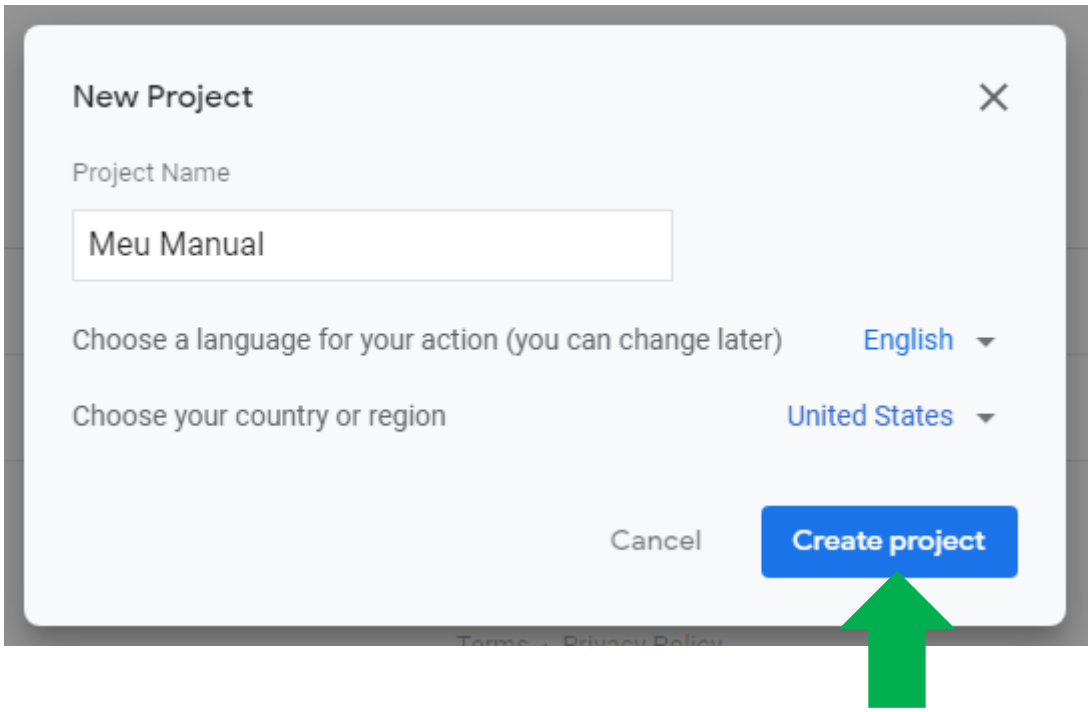


PASSO 3 (A criação da sua Nuvem)

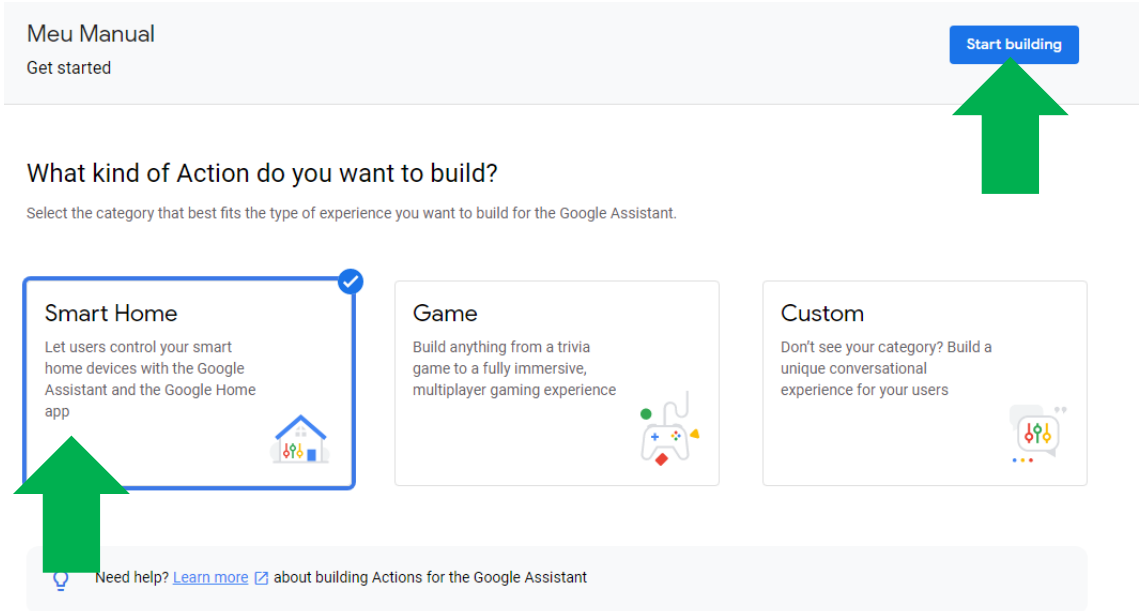
1) Vai em https://console.actions.google.com/



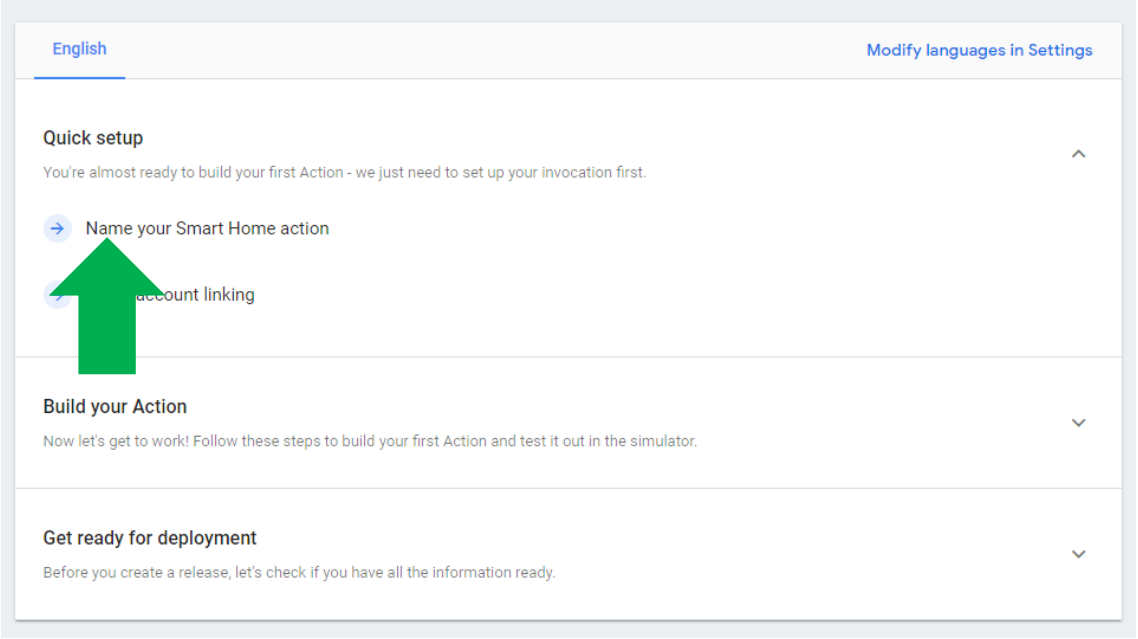
2) Clique em New Project e Crie com um nome a sua escolha.



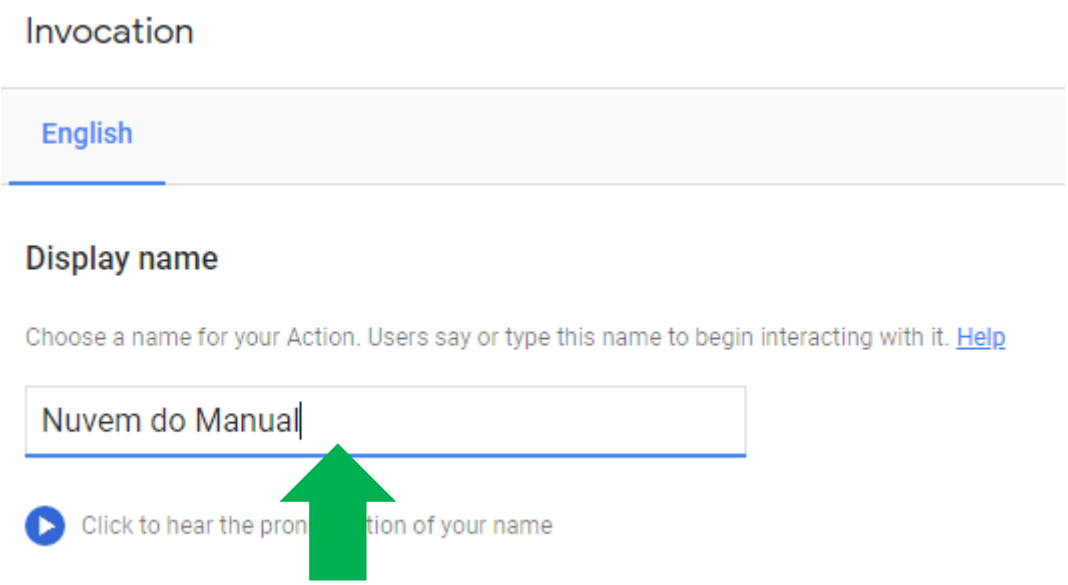
3) Clique em Smart Home e depois em Start Buildind



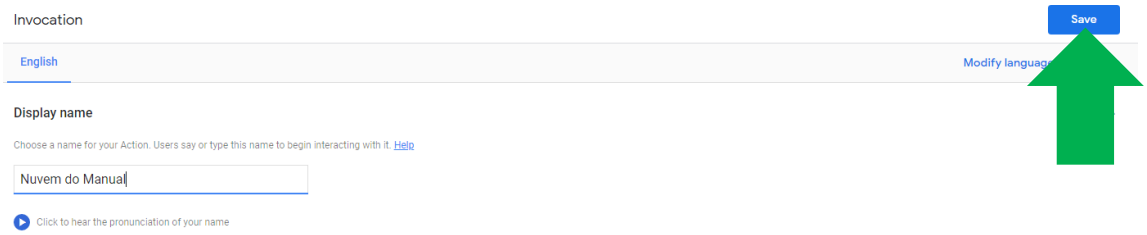
4) Agora em Quick setup, clique em Name your Smart Home action



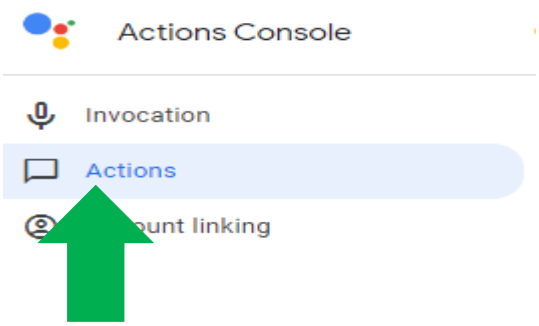
5) Crie um nome a sua escolha



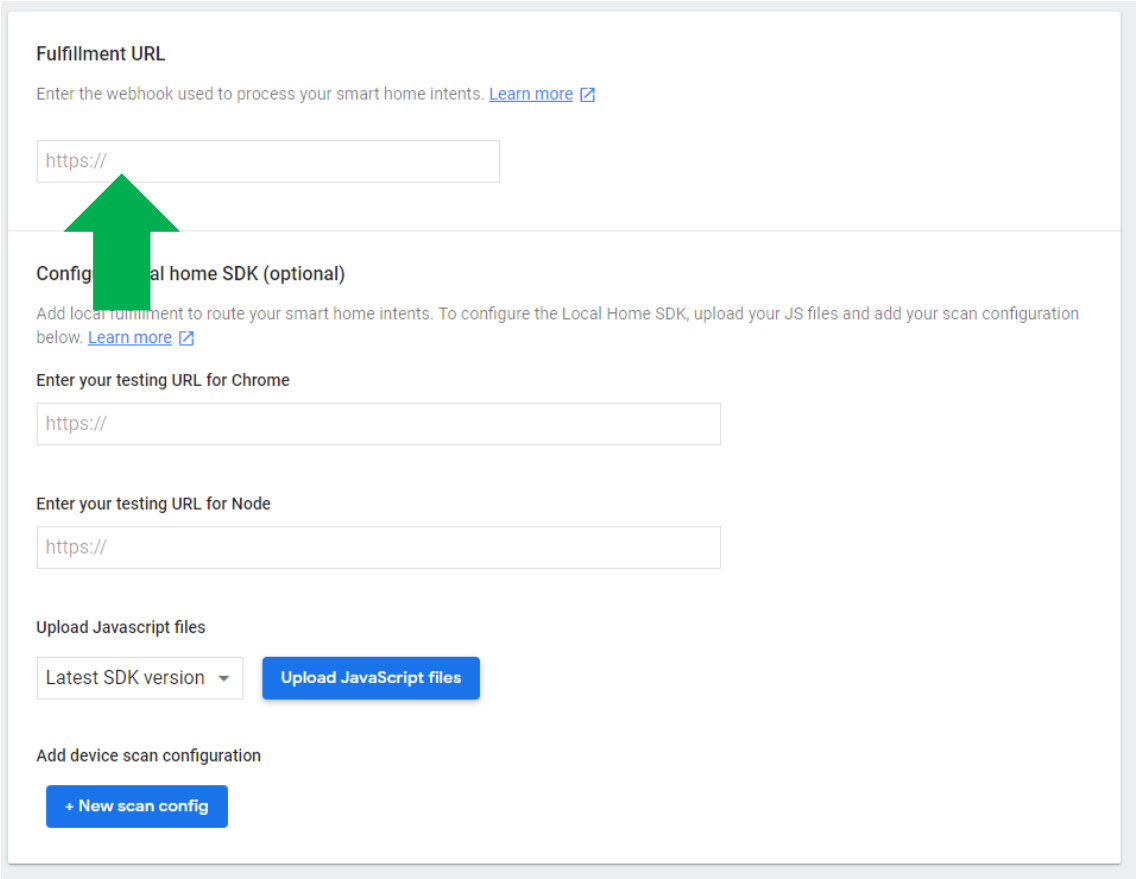
6) Clique em Salvar no canto superior direito.



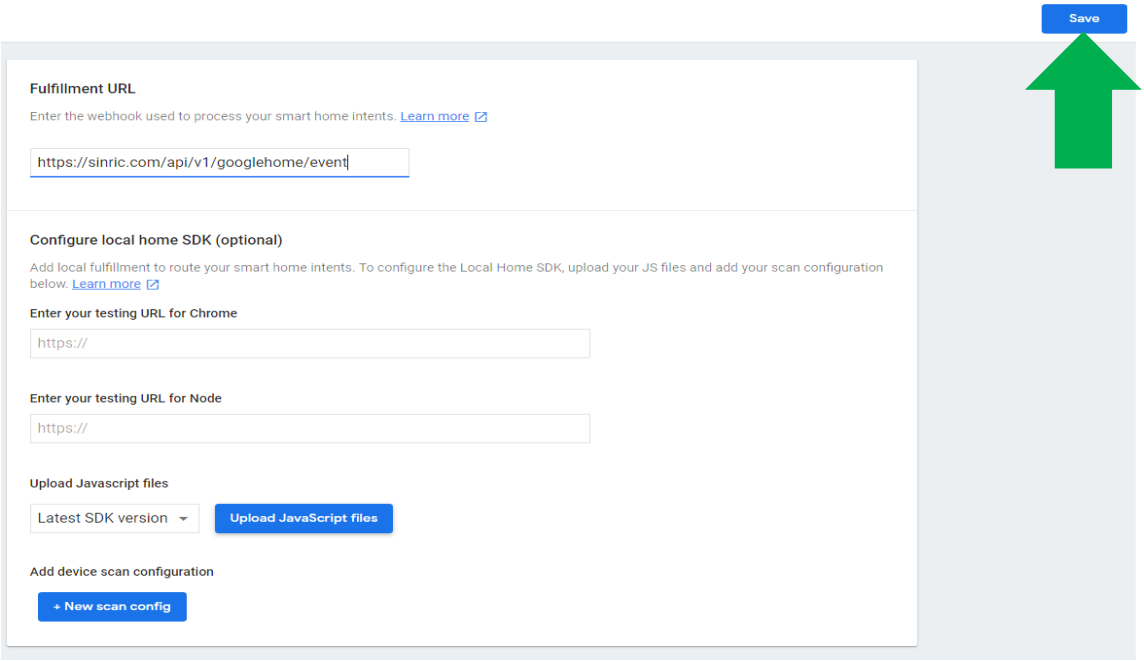
7) Clique em action do lado esquerdo da tela.



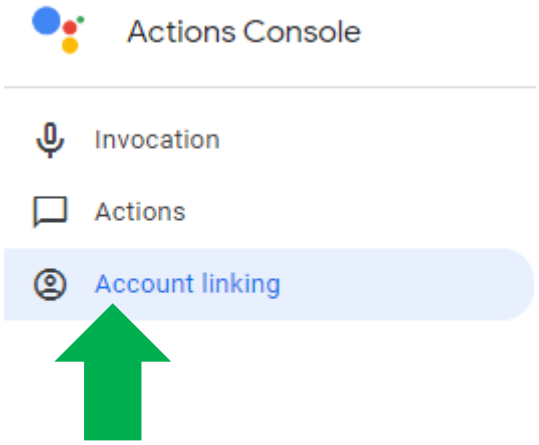
8) Cole esse link https://sinric.com/api/v1/googlehome/event em Fulfillment URL



9) Clique em Salvar no canto superior esquerdo.



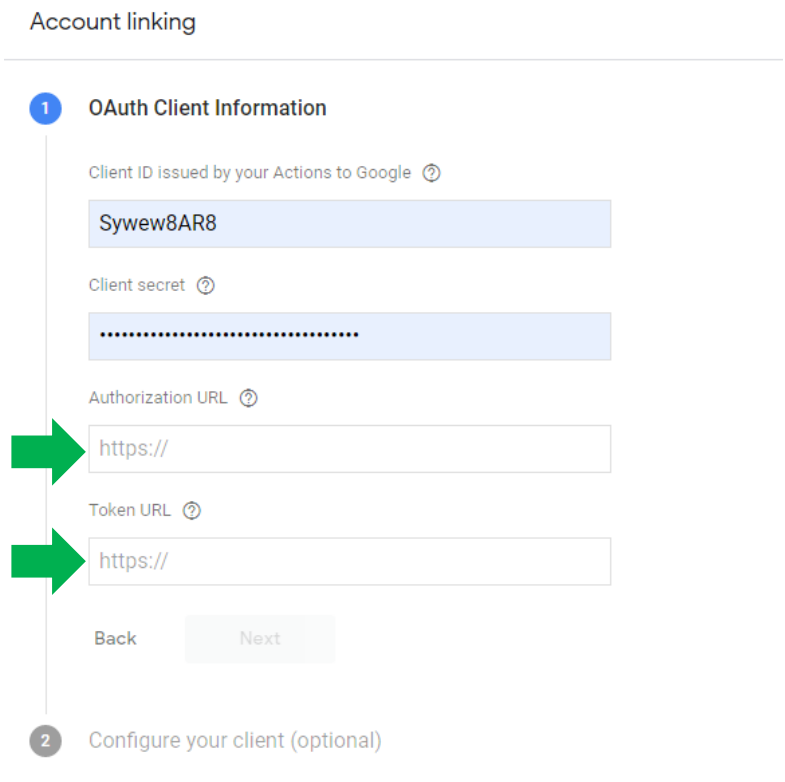
10) Clique em Account Linking



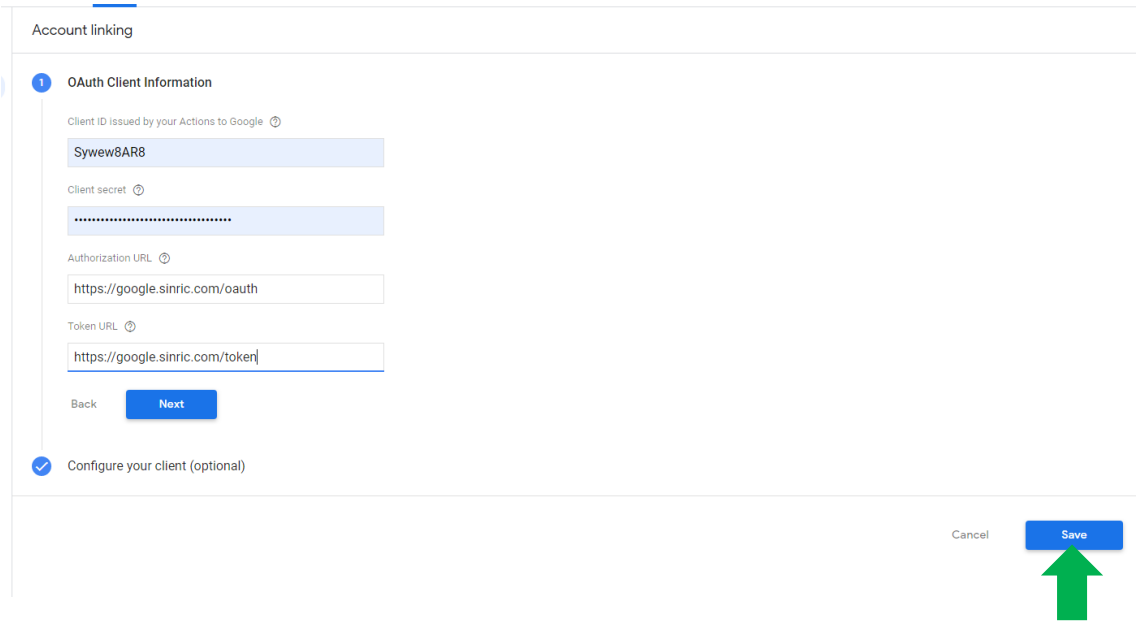
11) Na próxima imagem:

Em Autorization URL, coloque esse link: https://google.sinric.com/oauth

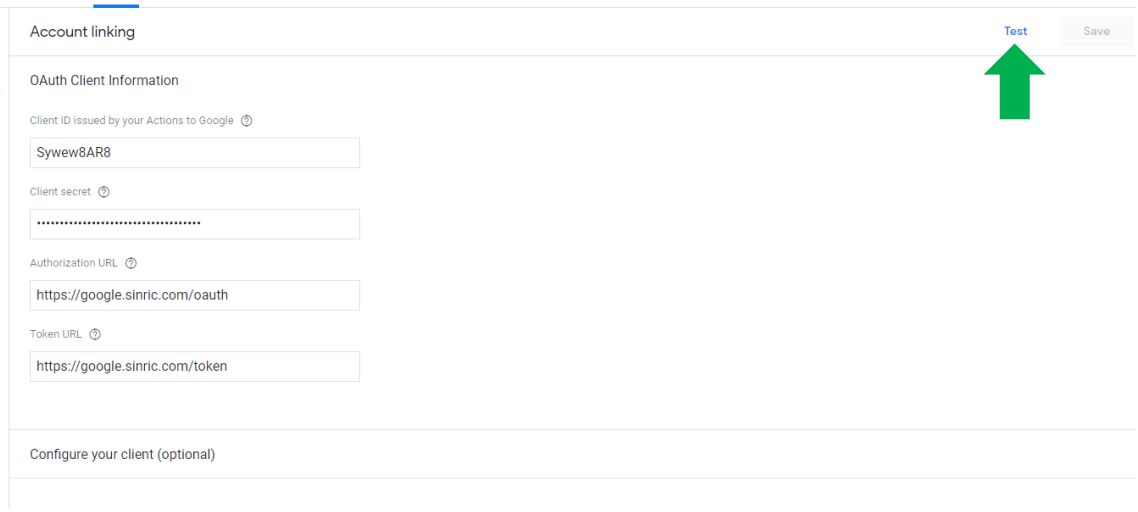
Em Token URL, esse outro link: https://google.sinric.com/token



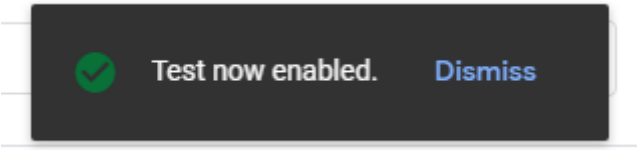
12) Clique em Salvar no canto inferior direito.



13) Agora clique em teste no canto superior direito.

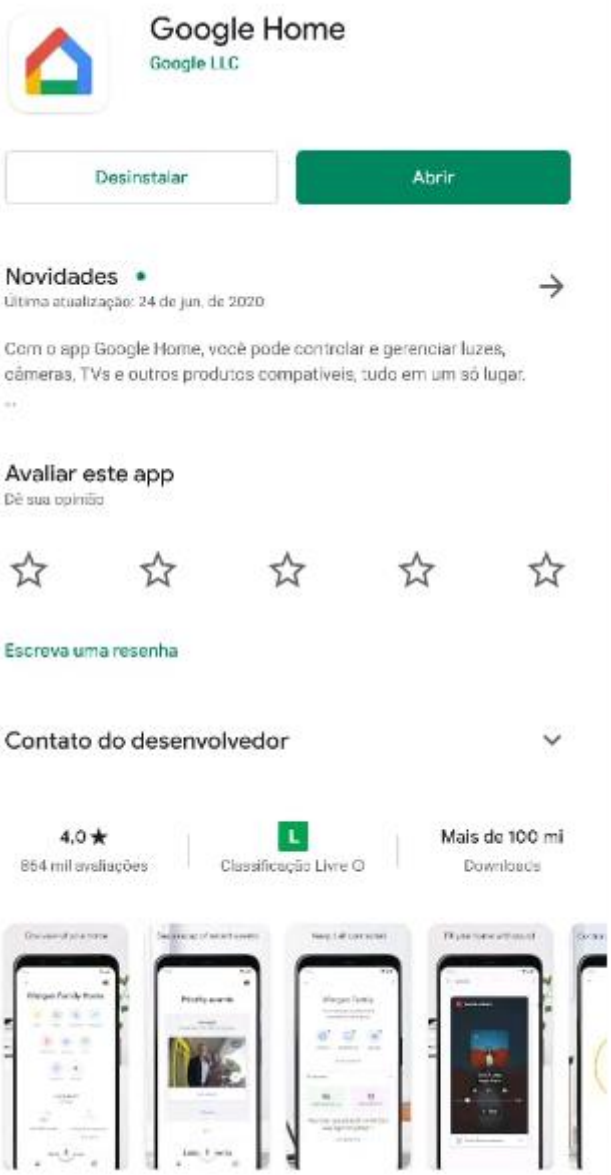


14) Caso esteja tudo certo vai aparecer uma mensagem assim:

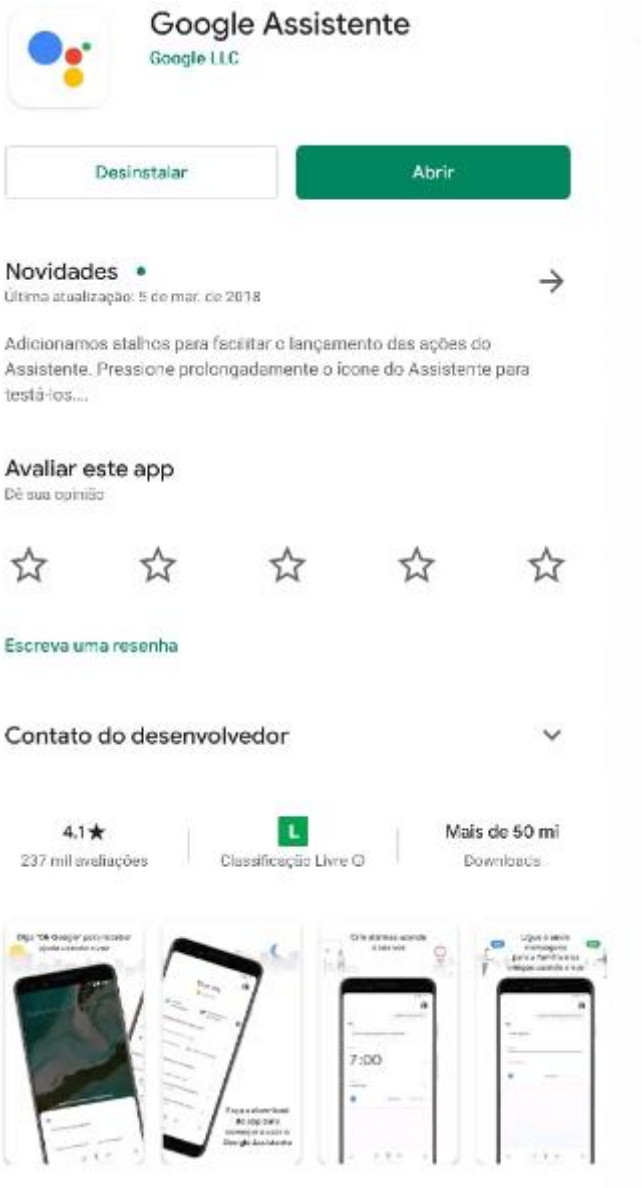


Passo 4 (Baixar os aplicativos no Smartphone e Configurar).

1) Vamos agora baixar o aplicativo Google Home no smartphone



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

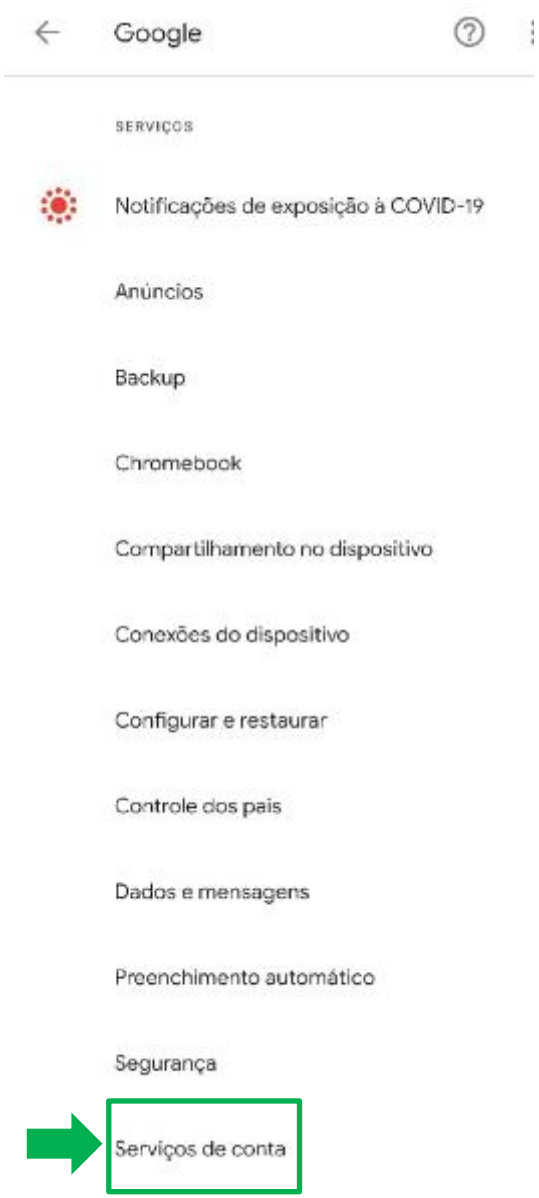


Vamos agora configurar o Google home:

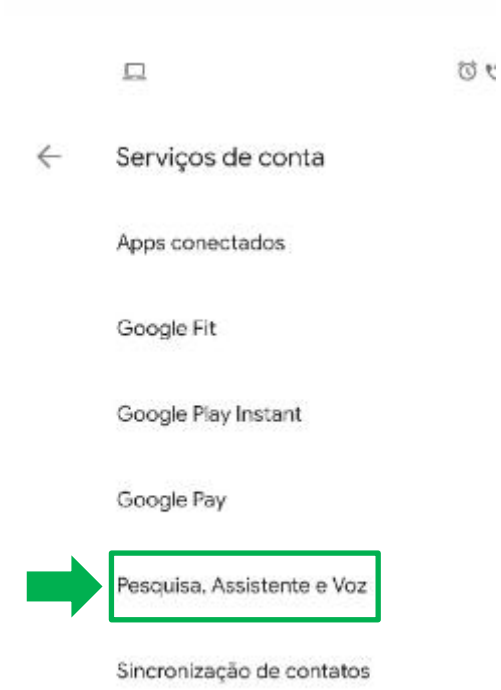
3) Abra configurações do seu smartphone.

4) Vai em Google – Serviços e preferencias dentro de configurações.

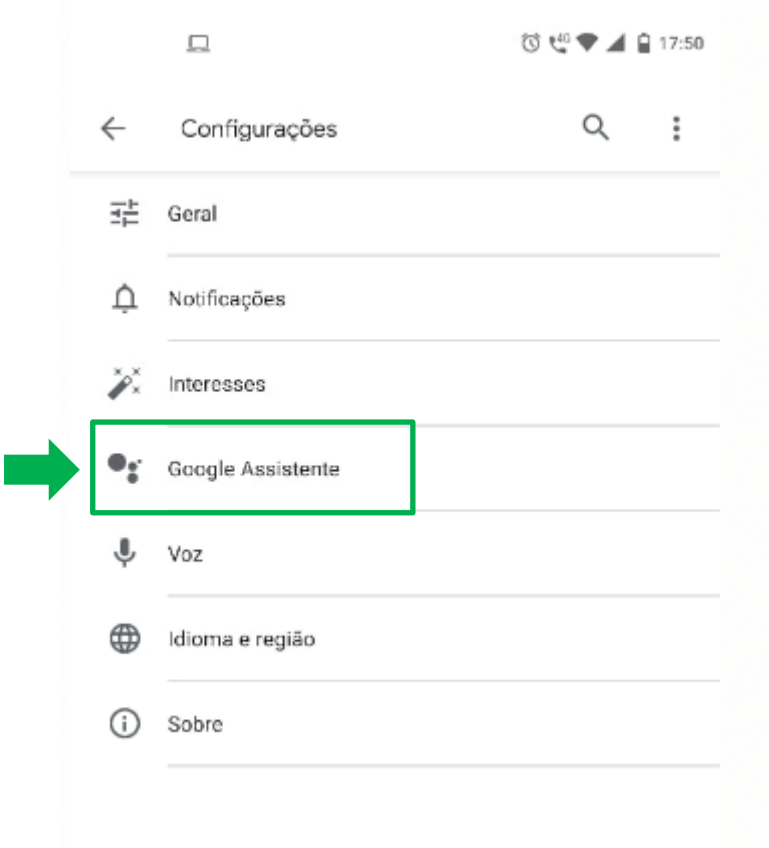
5) Vai em Seviços de Conta.



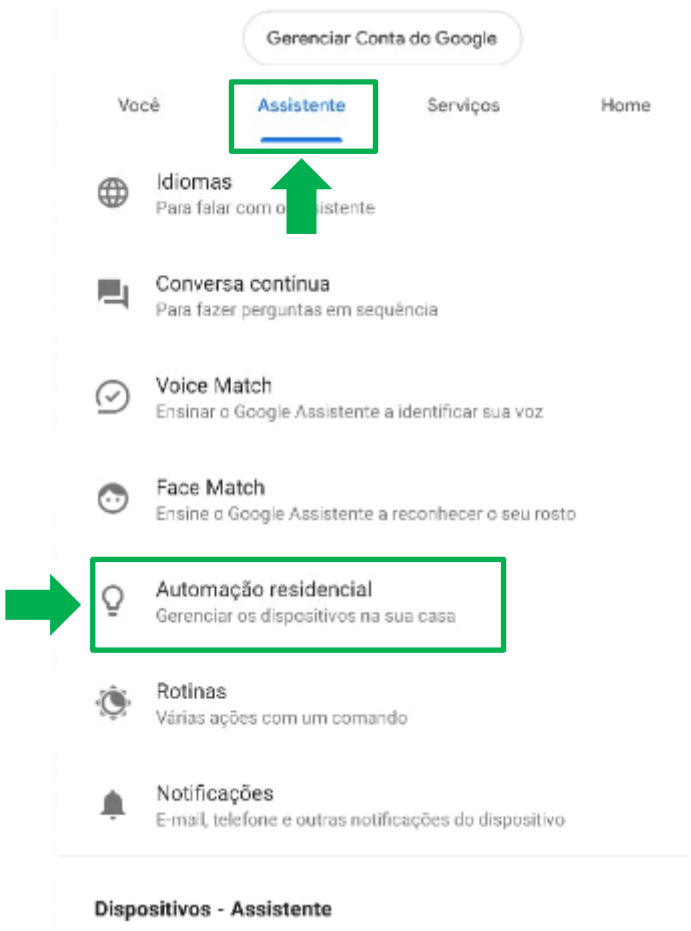
6) Vai Pesquisa Assistente e Voz.



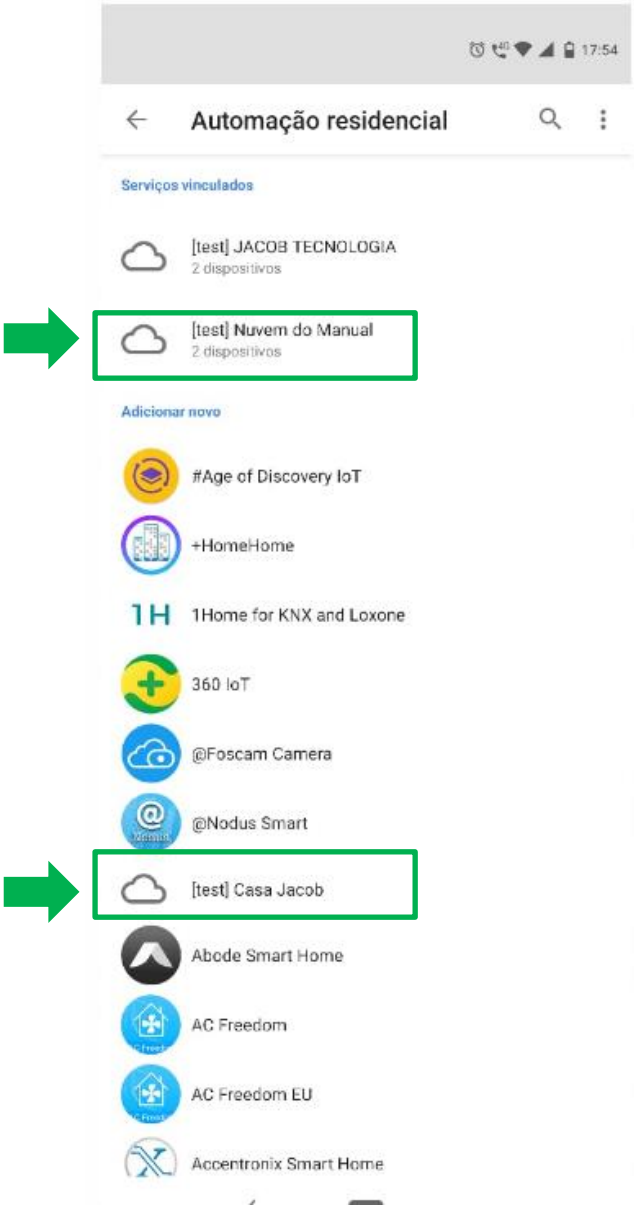
7) Vai em Google Assistente.



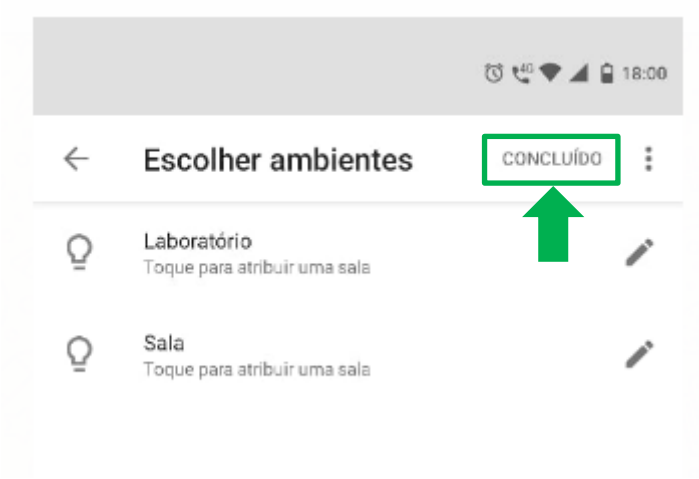
8) Clica em Assistente e depois em Automação Residencial



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 e só dedicar um tempo ai que vai dar tudo certo. Espero que tenha ajudado, pois deu trabalho pra fazer! Valeu!