# **INTRODUCCIÓN**

Esta práctica consiste en utilizar el asistente de voz de Google (Google Assistant) para controlar dispositivos conectados a un NodeMcu Esp8266, en este caso, para encender el Led de prueba que nos provee dicho dispositivo. Si bien puede parecer un objetivo trivial, puede escalarse para lograr algo mucho mas interesante, como encender las lámparas de una habitación en concreto de una casa o hasta un ventilador, todo con la voz.

Para llevar dicha práctica a cabo se usarán algunos servicios web y el asistente de Google de un dispositivo móvil.

# **CONCEPTOS CLAVE**

Primero es necesario explicar a grandes rasgos algunos conceptos y el funcionamiento general de los servicios web que usaremos:

# JSON

JavaScript Object Notation, es un formato del tipo “clave: valor” para el intercambio de datos, es muy similar a XML pero con una sintaxis más clara y acortada. JSON es un formato ampliamente usado en distintos lenguajes del lado del servidor para mandar datos al FrontEnd y poderlos procesar para mostrarlos al usuario.



# Webhooks



A muy grandes rasgos, los Webhooks son eventos que desencadenan acciones. Ahondando un poco más en el concepto, es una aplicación, o sitio web de terceros (el proveedor del Webhook) que envía una solicitud HTTP(POST) cuando ocurre un evento específico junto con mensaje (JSON), y un listener que recibe los Webhooks para así, realizar una acción predefinida.

Al operar los Webhooks bajo el concepto de “No me llames, yo te llamo si tengo algo para vos” (Event Reaction) se evita la necesidad de estar comprobando constantemente por cambios en un servidor, será el mismo servidor el que avise cuando haya cambios. Se podría decir que es una especie de suscripción entre servidores o de un cliente a un servidor, aunque para el último caso, existen los Websockets, que no serán tratados en el informe.

# Beebotte y API REST

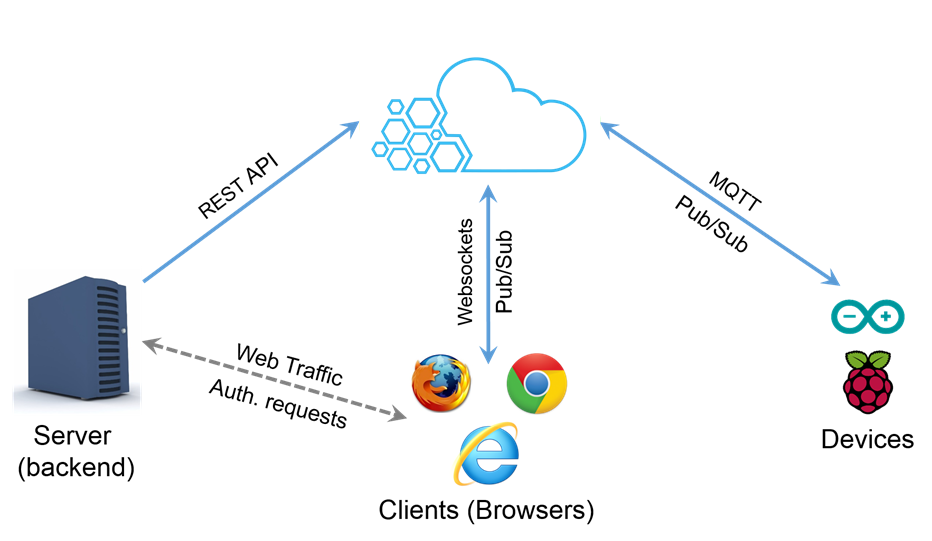


Basicamente Beebotte es un broker MQTT en la nube que fue diseñado específicamente para los desarrollos IoT, con una diferencia fundamental que lo diferencia de otros servicios similares y la que nos ayudará a realizar esta práctica de una manera mucho más sencilla, la API REST.

En pocas palabras, REST es cualquier interfaz entre sistemas que use HTTP para obtener datos o generar operaciones sobre esos datos en todos los formatos posibles, como XML y JSON.

Una API es un conjunto de definiciones y protocolos que se utiliza para desarrollar e integrar el software de las aplicaciones. API significa interfaz de programación de aplicaciones. Las API permiten que sus productos y servicios se comuniquen con otros, sin necesidad de saber cómo están implementados. A veces, las API se consideran como contratos, con documentación que representa un acuerdo entre las partes: si una de las partes envía una solicitud remota con cierta estructura en particular, esa misma estructura determinará cómo responderá el software de la otra parte.

Con esto, podemos darnos cuenta de que con Beebote podemos obtener datos a través de una solicitud HTTP.



# IFTTT



Es un servicio que permite crear eventos, estableciendo su disparador (if THIS) y la acción que se desencadena (then THAT). La plataforma conecta diferentes servicios online para automatizar todo tipo de tareas relacionadas con ellos, a cada conexión se le da el nombre de “applet” o “receta”. Puede decirse que IFTTT es intermediario entre más de 350 servicios web, entre ellos, Google Assistant y presta un servicio de Webhooks.

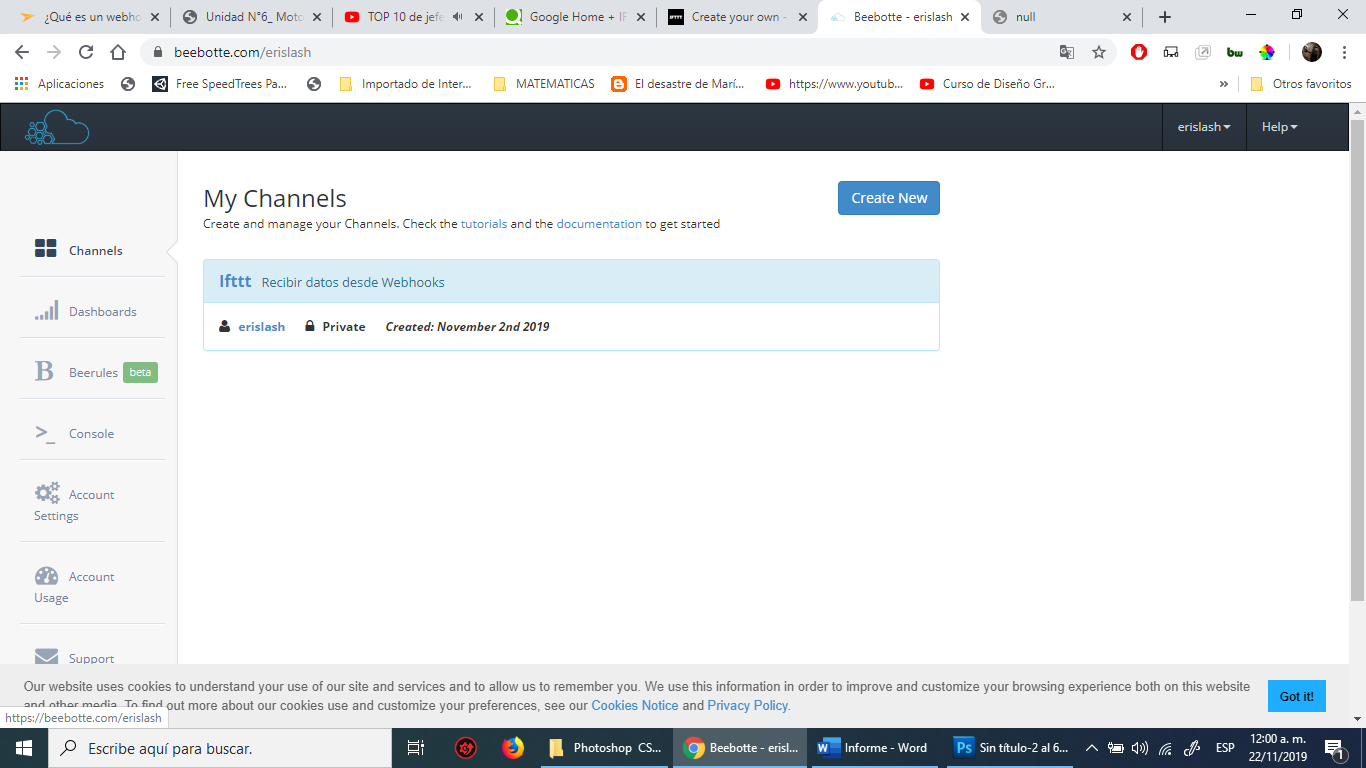
# **PRÁCTICA**



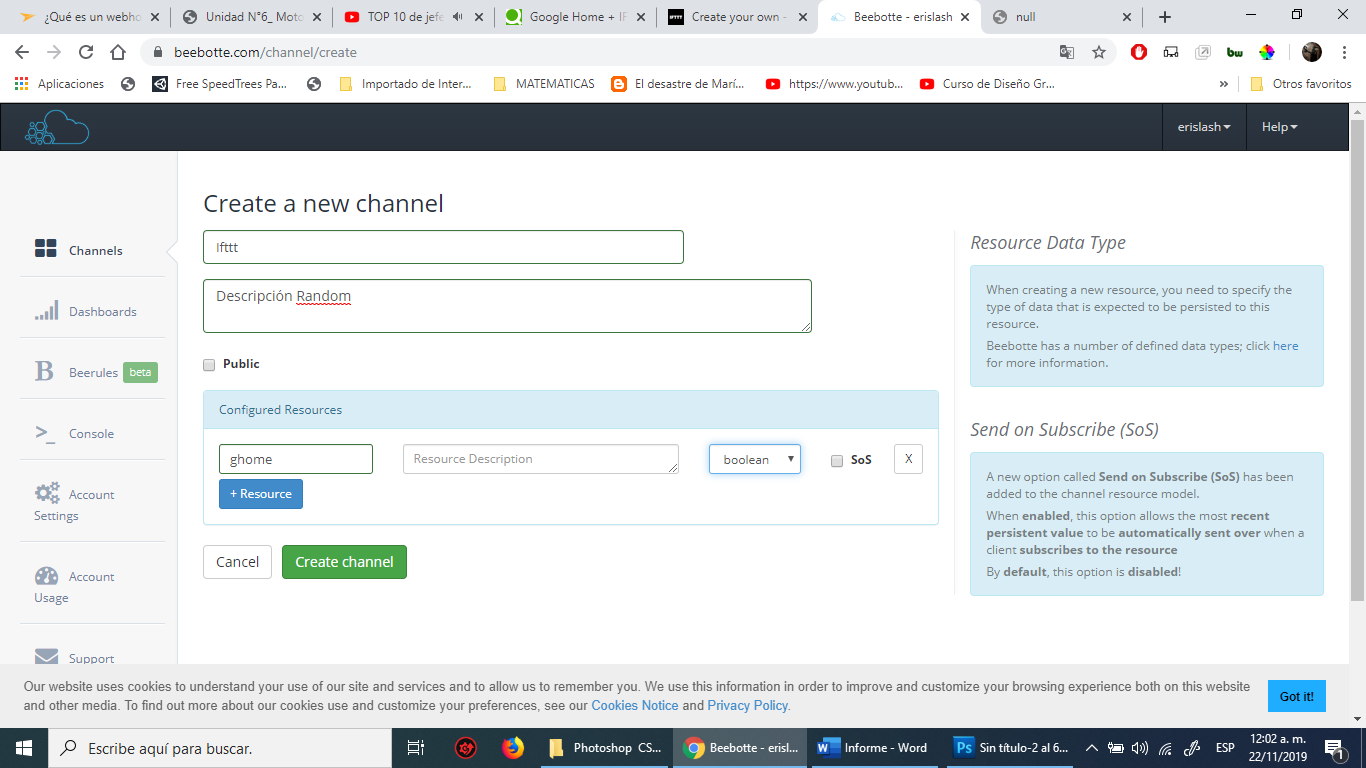
# Configuración del Broker MQTT

Lo primero será crearse una cuenta en Beebotte (<https://beebotte.com/>) ingresando un nombre de usuario, un E-mail y una contraseña. Luego de haber confirmado nuestra cuenta a través del correo que nos enviarán ya vamos a poder ingresar al Dashboard de Beebotte.

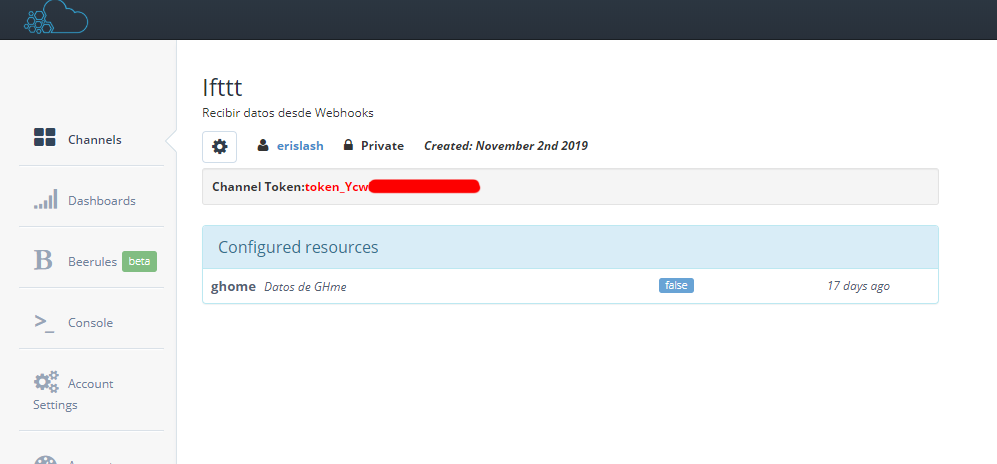
Una vez hecho esto, nos vamos al apartado de “Channels” y presionamos “Create New”.



Completamos con los datos requieridos y pulsamos “Create Channel”. El nombre del canal será Ifttt y la dirección en la que publicaremos los datos que enviaremos el Webhook será ghome.



Una vez completado el paso anterior podremos ingresar al canal que creamos. El “Channel Token” nos servirá para suscribirnos con el Node.



Por ahora hasta acá dejamos el trabajo en Beebotte y pasar a Sketch, para programar el Node ESP8266.

# PROGRAMACIÓN DEL NODE

//LIBRERÍAS A USAR

#include <ESP8266WiFi.h>

#include <PubSubClient.h>

#include <ArduinoJson.h>

//CONFIGURACIÓNES INICIALES

#define led  2

#define ssid  "RED"

#define wifiPass "12345678"

#define mqttServer "mqtt.beebotte.com"

#define mqttPort 1883

//TOKEN QUE NOS PROVEE BEEBOTTE

#define mqttUser "token\_YcwMxxxxxxxxxxxxxx"

#define subTopic "Ifttt/ghome" //DIRECCIÓN QUE ASIGNAMOS ANTERIORMENTE

WiFiClient espClient;

PubSubClient client(espClient);

long lastAttempt = 0;

void setup(){

  Serial.begin(115200);

  pinMode(led, OUTPUT);

  ConnectWifi(ssid, wifiPass);

  client.setServer(mqttServer, mqttPort);

  Serial.println("Conectando a MQTT");

  while(!client.connected()){

    Serial.println(".");

    if(client.connect("NodeLuz", mqttUser, "")){

      Serial.println("Conectado");

      client.subscribe(subTopic);

      Serial.println("Suscripto al Tema: ");

      Serial.println(subTopic);

      client.setCallback(CallMQTT);

    }else{

      Serial.print("No se pudo Conectar al Broker MQTT");

      Serial.print(client.state());

      delay(2000);

    }

  }

 client.setCallback(CallMQTT);

}

void loop(){

//RECONEXIÓN AL SERVIDOR

  if (!client.connected()) {

    long now = millis();

    if (now - lastAttempt > 5000) {

      lastReconnectAttempt = now;

      if (reconnect()) {

        lastAttempt = 0;

      }

    }

  } else {

    client.loop();

  }

}

//FUNCIÓN PARA CONECTARSE A UNA RED WIFI

void ConnectWifi(String p\_ssid, String p\_password){

  WiFi.mode(WIFI\_OFF);

  delay(1000);

  WiFi.mode(WIFI\_STA);

  WiFi.begin(p\_ssid, p\_password);

  Serial.println();

  Serial.print("Intentando Conectar a la Red: ");

  Serial.println(p\_ssid);

  while (WiFi.status() != WL\_CONNECTED) {

    delay(500);

    Serial.print(".");

  }

  Serial.println("Conectado a la Red");

}

/\*CALLBACK, SE EJECUTARÁ AL RECIBIR UN MENSAJE DEL TEMA

AL QUE NOS SUSCRIBIMOS\*/

void CallMQTT(char\* topic, byte\* payload, unsigned int length) {

  String message = "";

  Serial.print("Ha llegado un mensaje desde el tema: ");

  Serial.println(topic);

  Serial.print("Mensaje:");

  for (int i = 0; i < length; i++) {

    message += (char)payload[i];

  }

  Serial.println(message);

/\*DESERIALIZACIÓN DEL JSON QUE NOS LLEGA A TRAVÉS DEL BROKER,

ENVIADO POR EL WEBHOOK\*/

  StaticJsonDocument<256> doc;

  DeserializationError error = deserializeJson(doc, message);

  if(error) {

    Serial.print(F("deserializeJson() failed: "));

    Serial.println(error.c\_str());

    return;

  }

//EXTRAEMOS EL DATO GUARDADO EN LA KEY "data"

  boolean action = doc["data"];

  Serial.println();

  Serial.println("/////////////////////////////////////////////////////////////////");

/\*SI EL MENSAJE QUE RECIBIMOS ES "true" ENCENDEMOS EL LED

E INFORMAMOS QUE SE ENCENDIÓ A TRAVÉS DE OTRA DIRECCIÓN

A LA QUE SE SUSCRIBIRÁ LA APLICACIÓN IoT "MQTT Panel"\*/

  if(action){

    digitalWrite(led, LOW);

    client.publish("Ifttt/nodeStatus", "{\"data\":true}");

  }

//DE RECIBIR UN "false", SE APAGARÁ E INFORMARÁ

  else if(!action){

    digitalWrite(led, HIGH);

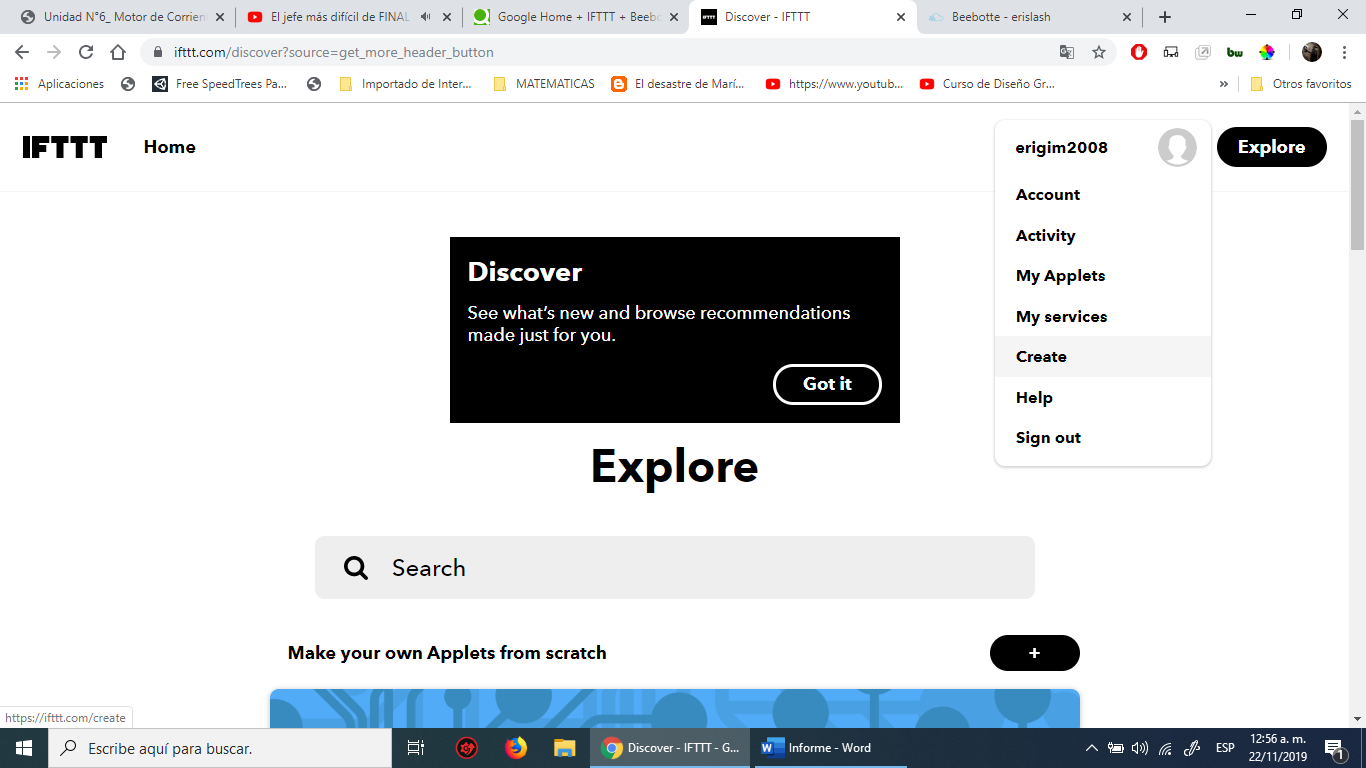
    client.publish("Ifttt/nodeStatus", "{\"data\":false}");

  }

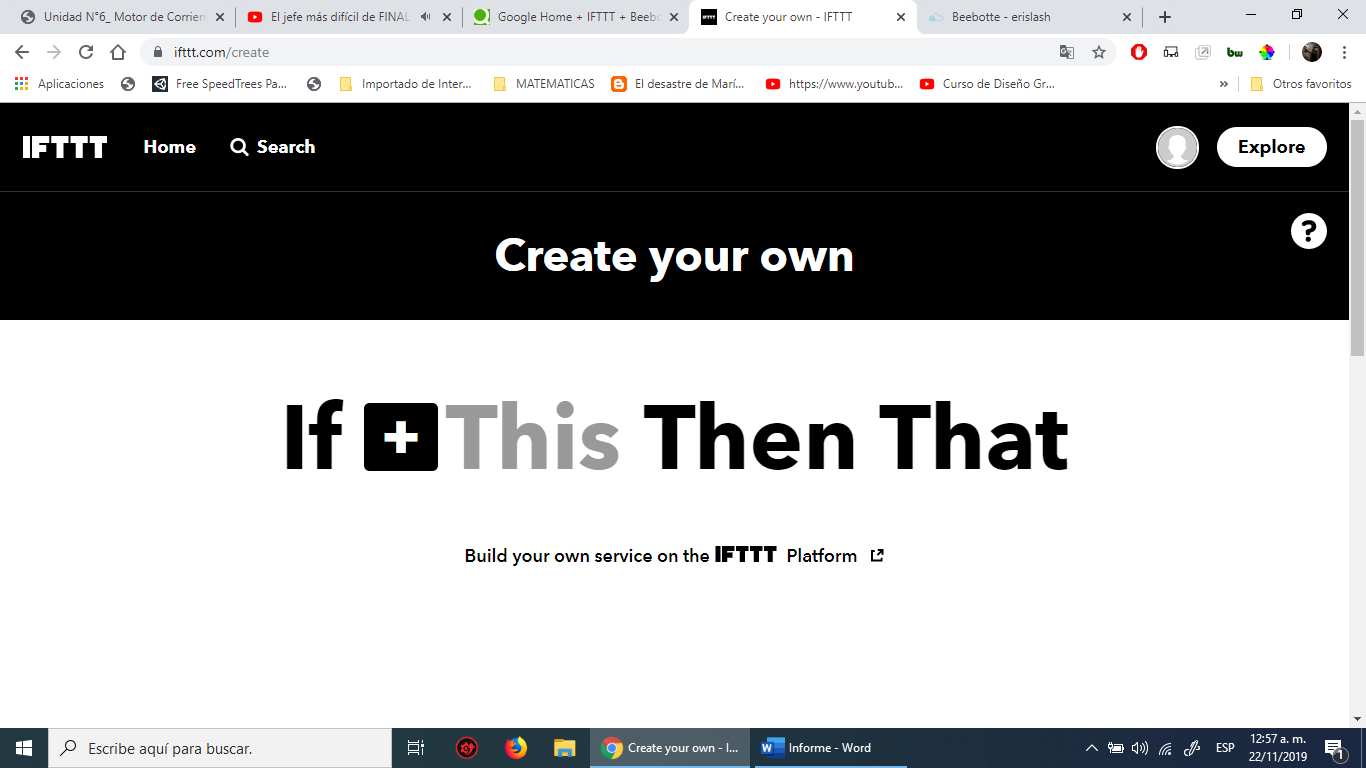
}

# CONFIGURACIÓN DE IFTTT

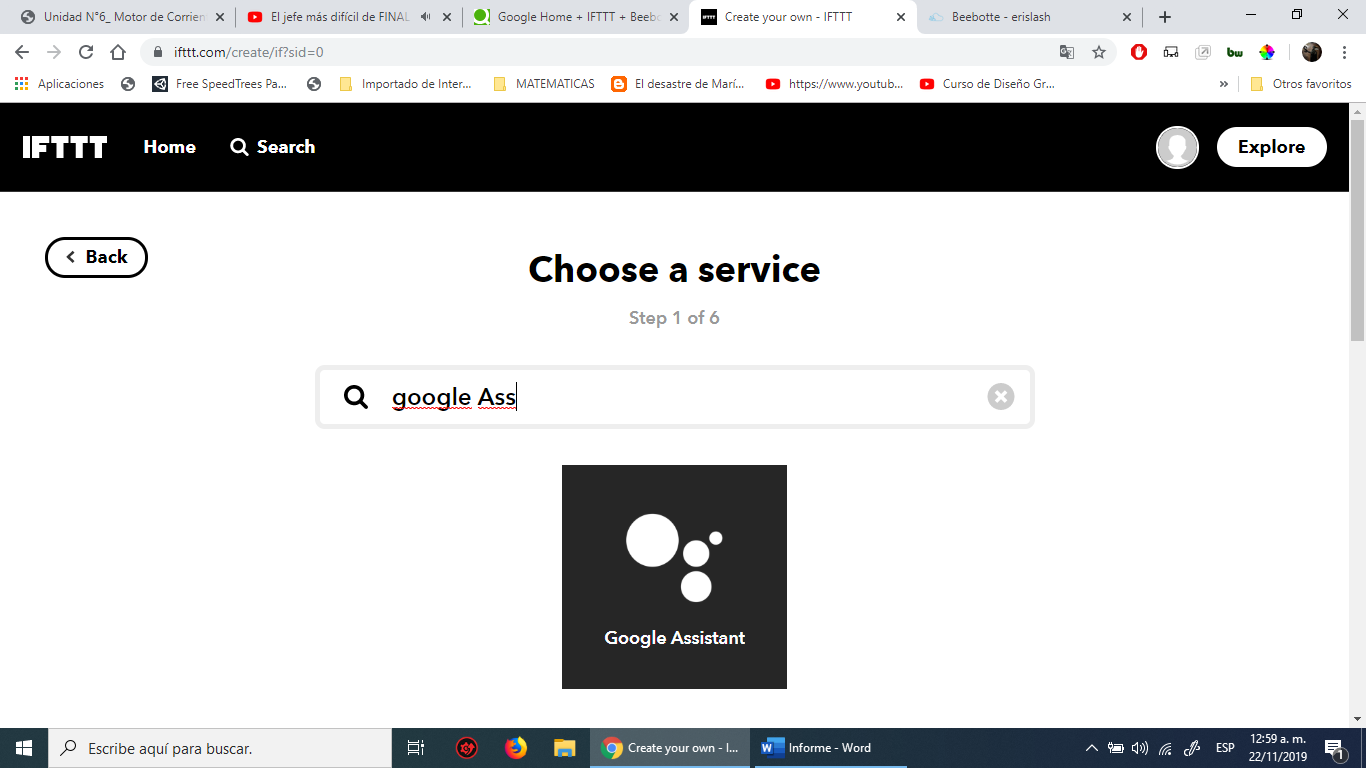
Hay que crearse una cuenta en la plataforma IFTTT (<https://ifttt.com/>) usando nuestra cuenta de Google. Una vez hecho esto, presionaremos en nuestra imagen de perfil en la esquina derecha superior y le damos a “Create” para comenzar a crear nuestra applet o receta.



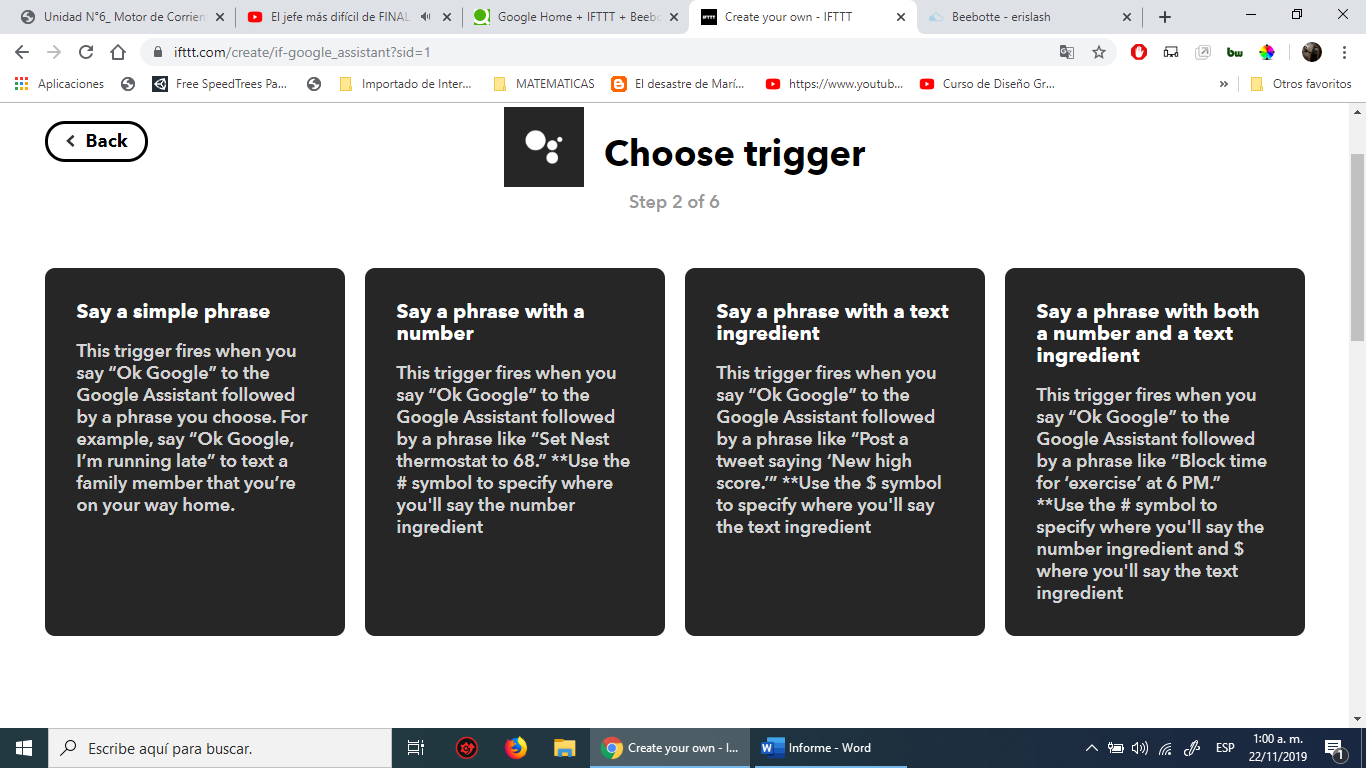
Eso nos llevará a una página así (o algo así, la interfaz cambia cada rato).



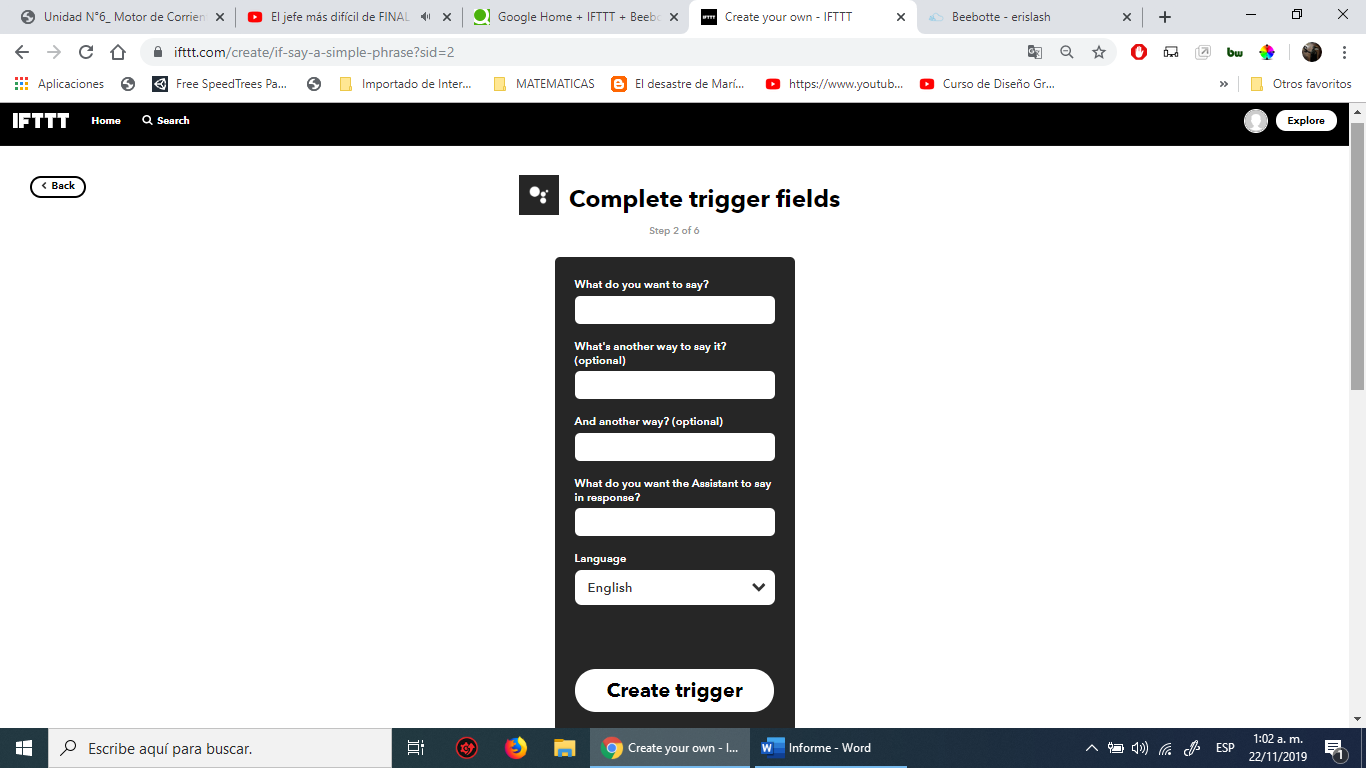
Presionamos en el signo “+” y buscamos Google Assistant entre la gran cantidad de servicios.



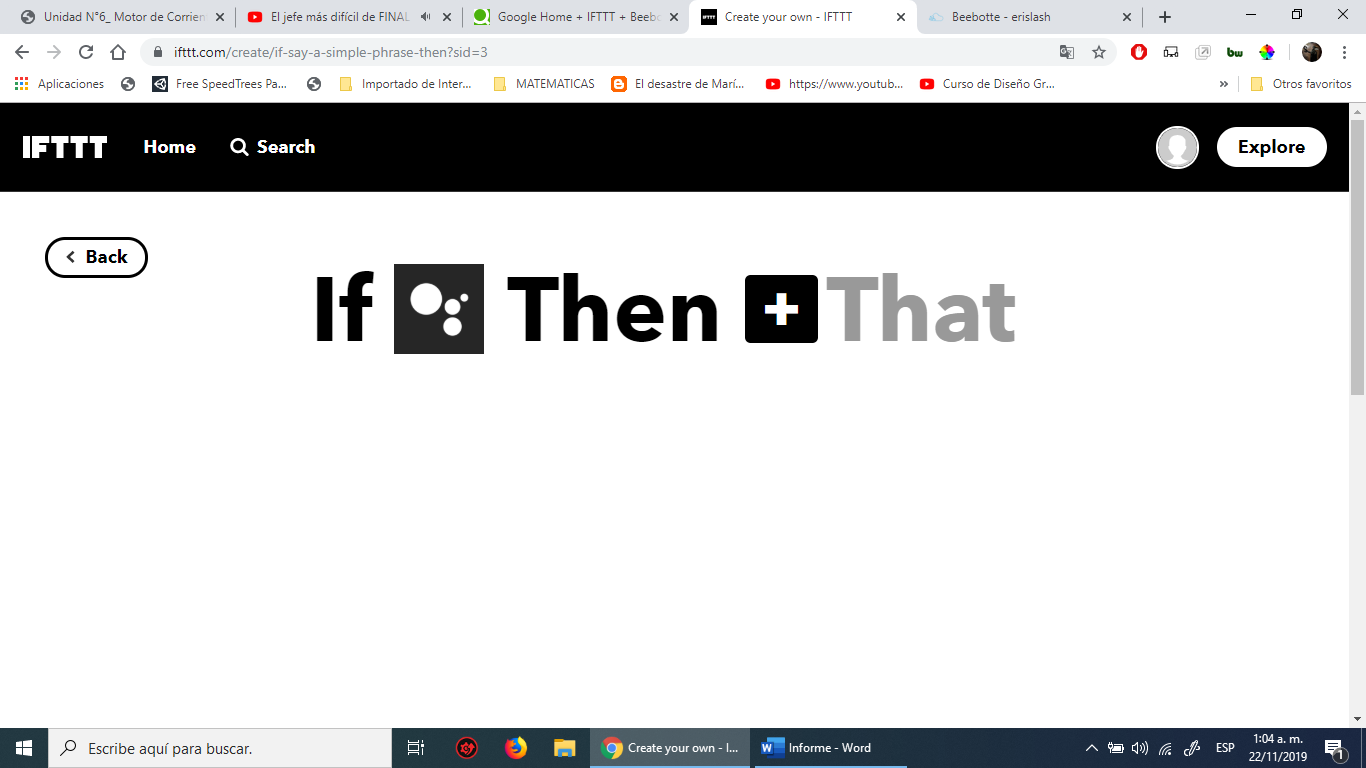
Una vez seleccionado, podremos elegir la acción que enviará un Webhook. En este caso, “Say a simple phrase”.



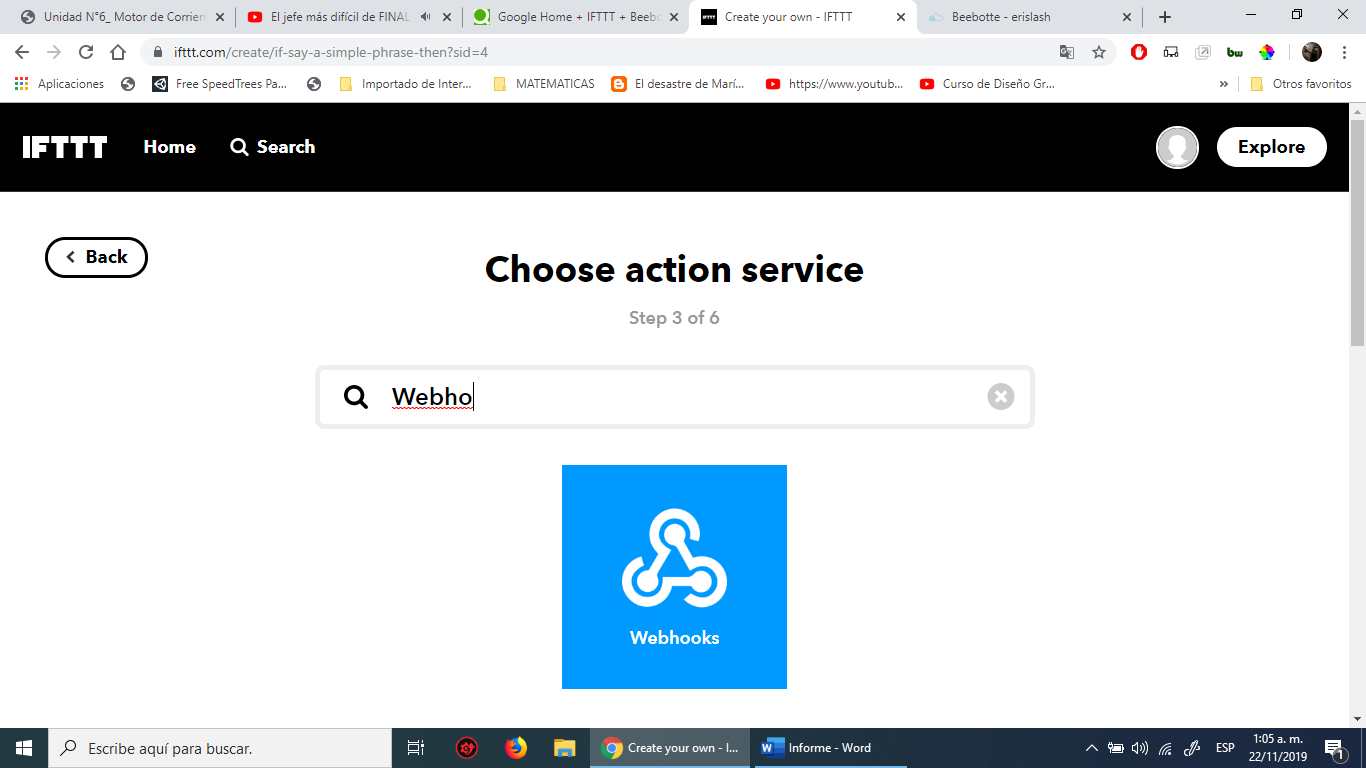
Nos aparecerá otra pantalla como esta. La cual completaremos con las frases que queremos que envíen el Webhook (se debe hacer una receta para prender y otra para apagar) y la respuesta que nos dará el asistente, además del idioma que usaremos.



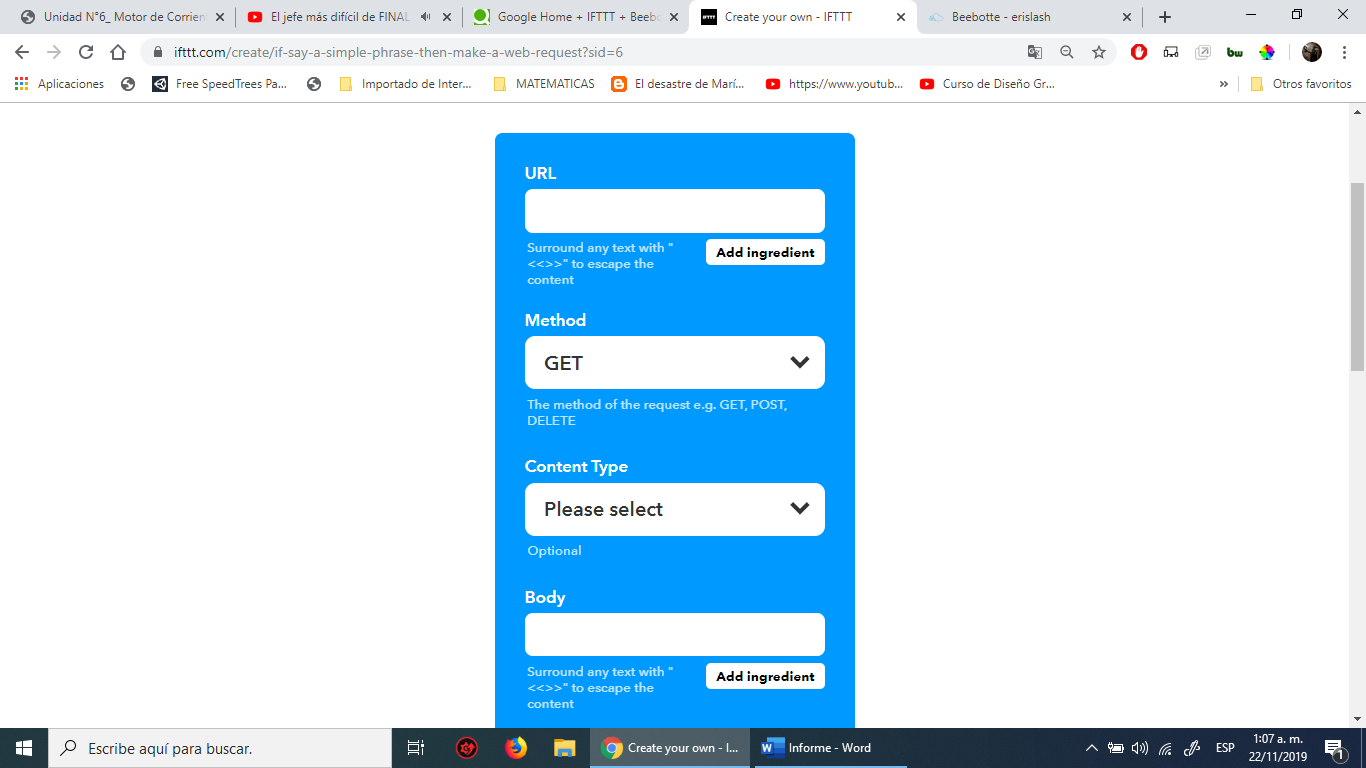
Y ahora solo queda enviar el Webhook, los pasos son casi los mismos de recién.



Buscamos Webhooks.



Y seleccionamos la única opción “Make a web request” y completamos los datos.



**URL**: <https://api.beebotte.com/v1/data/publish/Ifttt/ghome?token=token_Ycwxxxxxxxxxxxxx>

Este link es la API REST de beebote, luego de publish se coloca el nombre del canal al que enviaremos los datos y después de eso, el nombre de recurso, para terminar, incluyendo el token que ya pusimos en la programación del node. Para determinar el link necesario de una forma más fácil puede usarse la página de Beebotte <https://beebotte.com/api/play>.

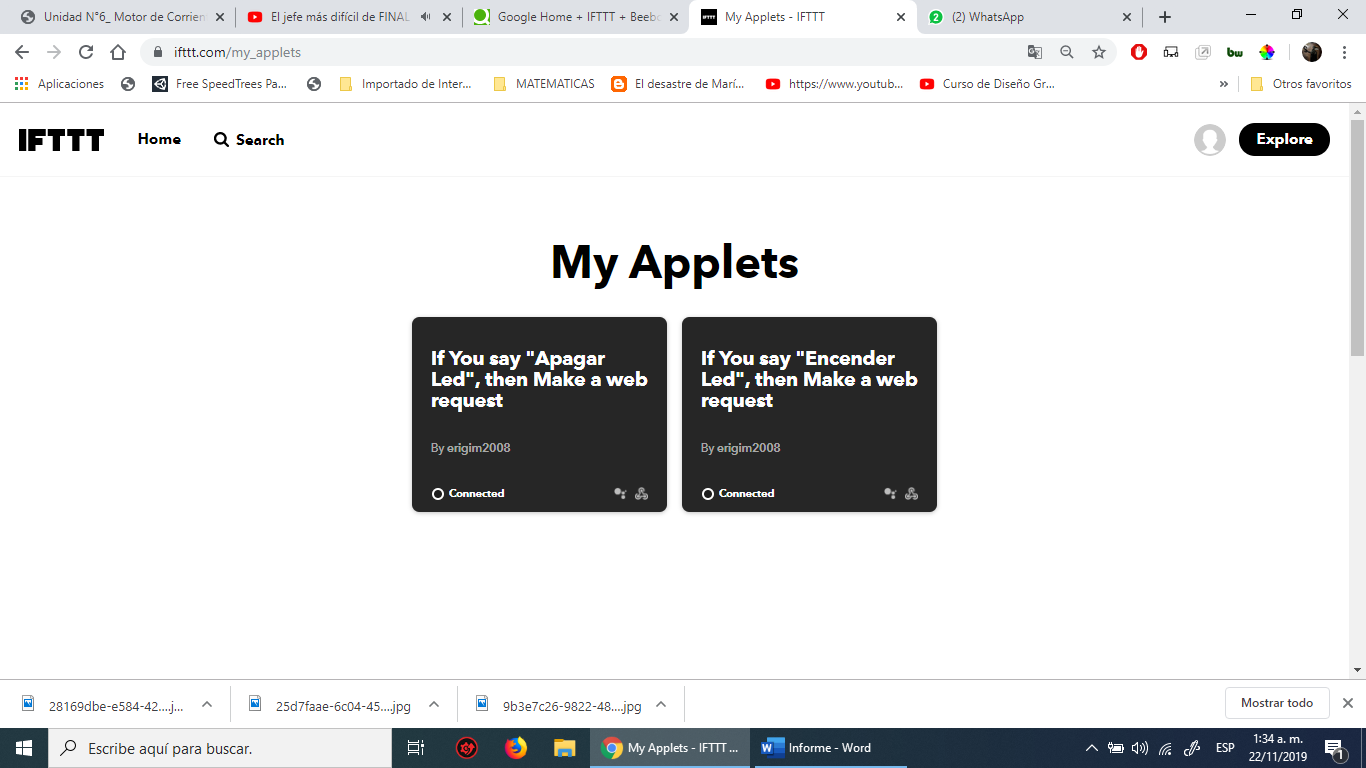
**Method**: POST

**Content Type**: application/json

**Body**:

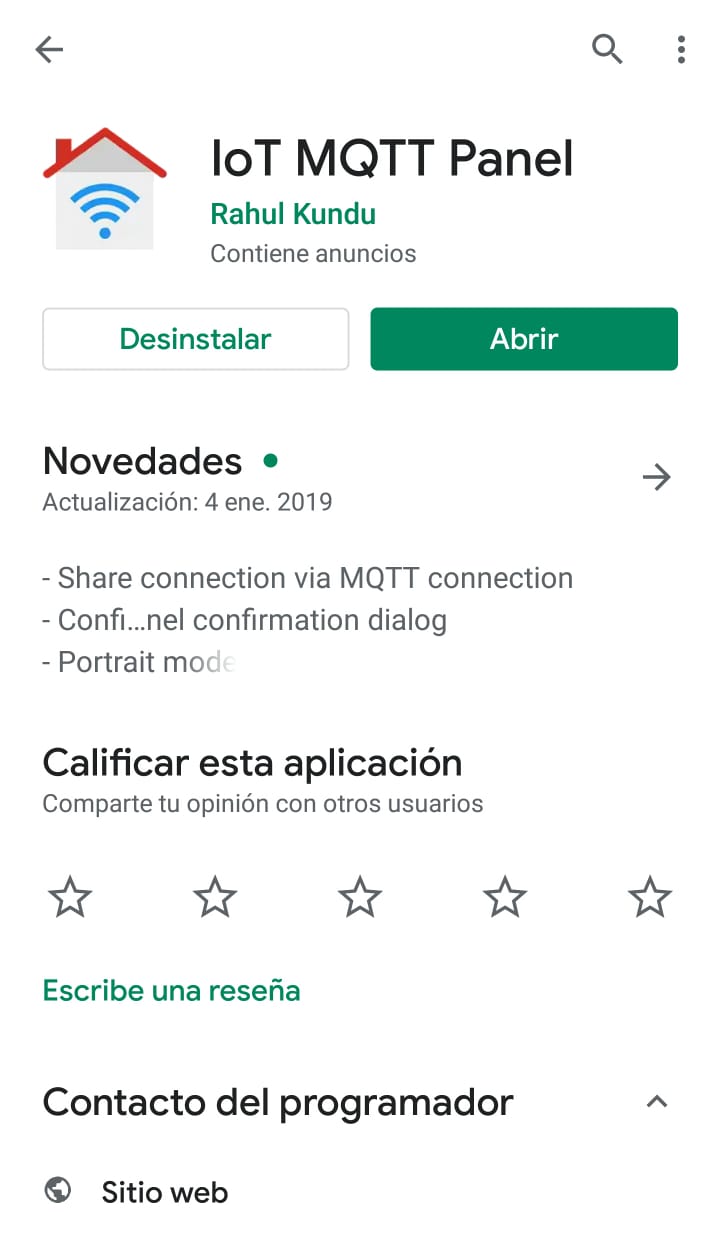
* {“data”:true} (si queremos encender el led)
* {“data”:false} (si queremos apagar el led)

No es necesario usar un valor booleano, tranquilamente podría haber usado una palabra o frase, lo importante es la comparación en el node.

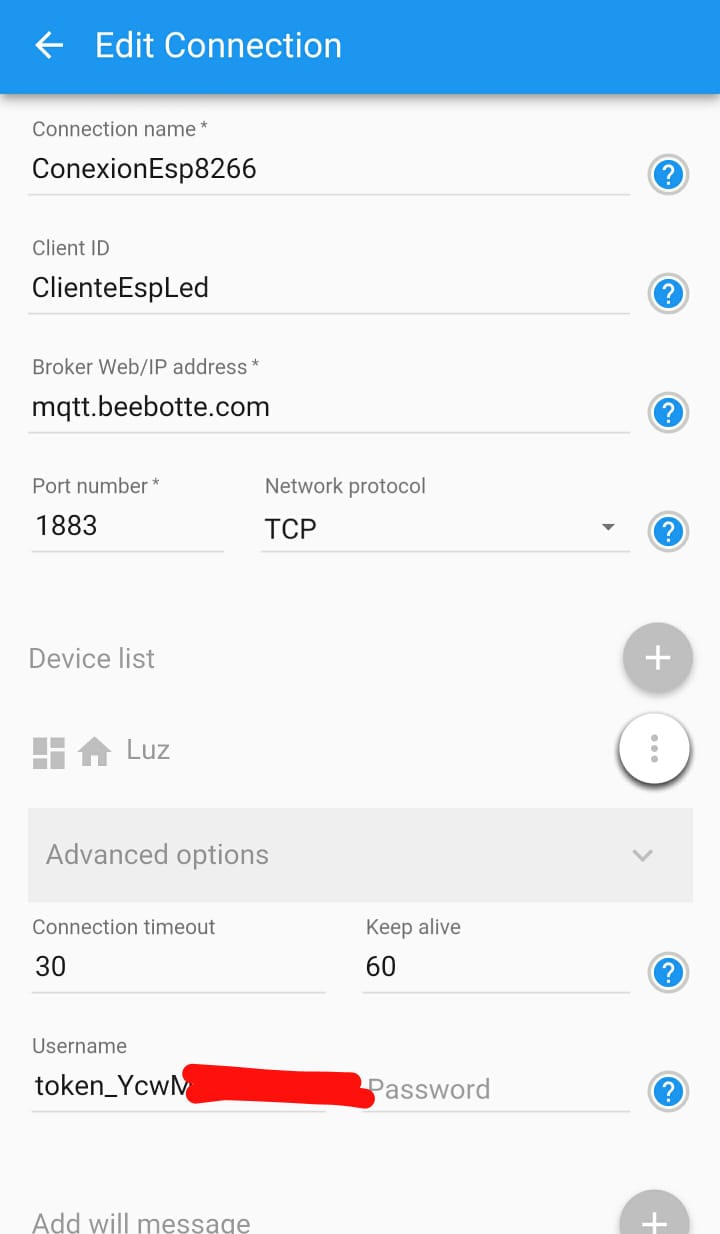
Vuelvo a remarcar que se deben hacer dos recetas, una para encender el led y otra para apagarlo. 

# INTEGRACIÓN DE UN DASHBOARD EN EL MÓVIL

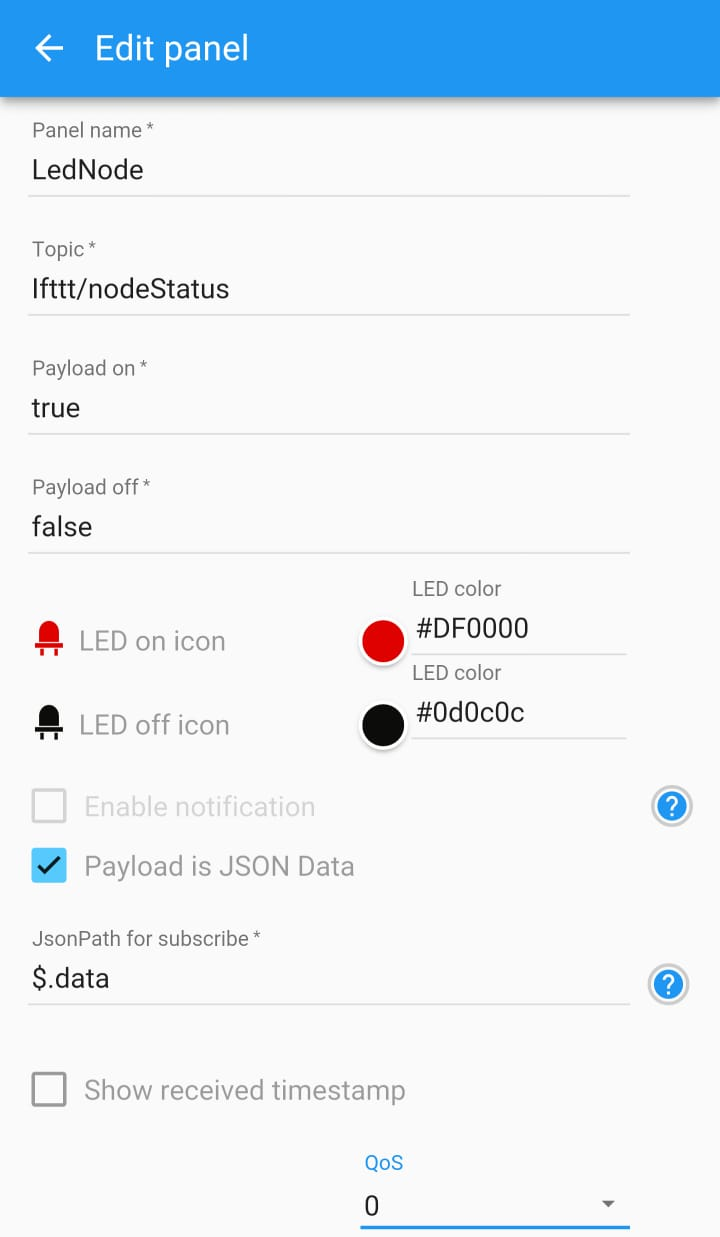
Lo primero es descargar una aplicación dedicada a IoT. En este caso usamos “Iot MQTT Panel”.



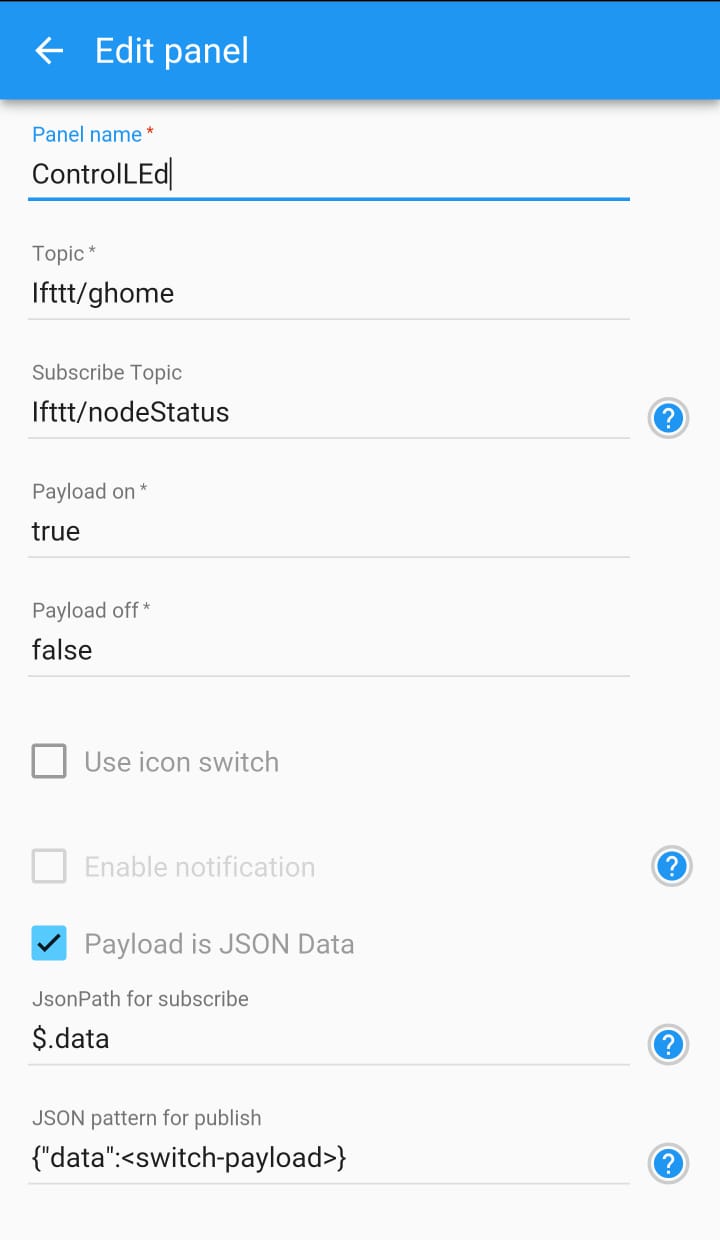
La iniciamos y añadimos una nueva conexión con el botón “+”.



Una vez conectados, añadimos nuevos widgets al panel, primero un indicador del led del node.



Y por último un switch, para además de controlar el led con Google Assistant, podremos controlarlo con esta app.



Con todos estos pasos, ya tenemos nuestra practica hecha. Como dije al principio, es simplemente un ejemplo, pero puede usarse para controlar dispositivos mucho mas grandes que un simple led. Las herramientas y conceptos necesarios están cubiertos a grandes rasgos en este informe y es el pie inicial para comenzar a investigar sobra cada uno de esos conceptos ya que cada uno es un mundo de información.