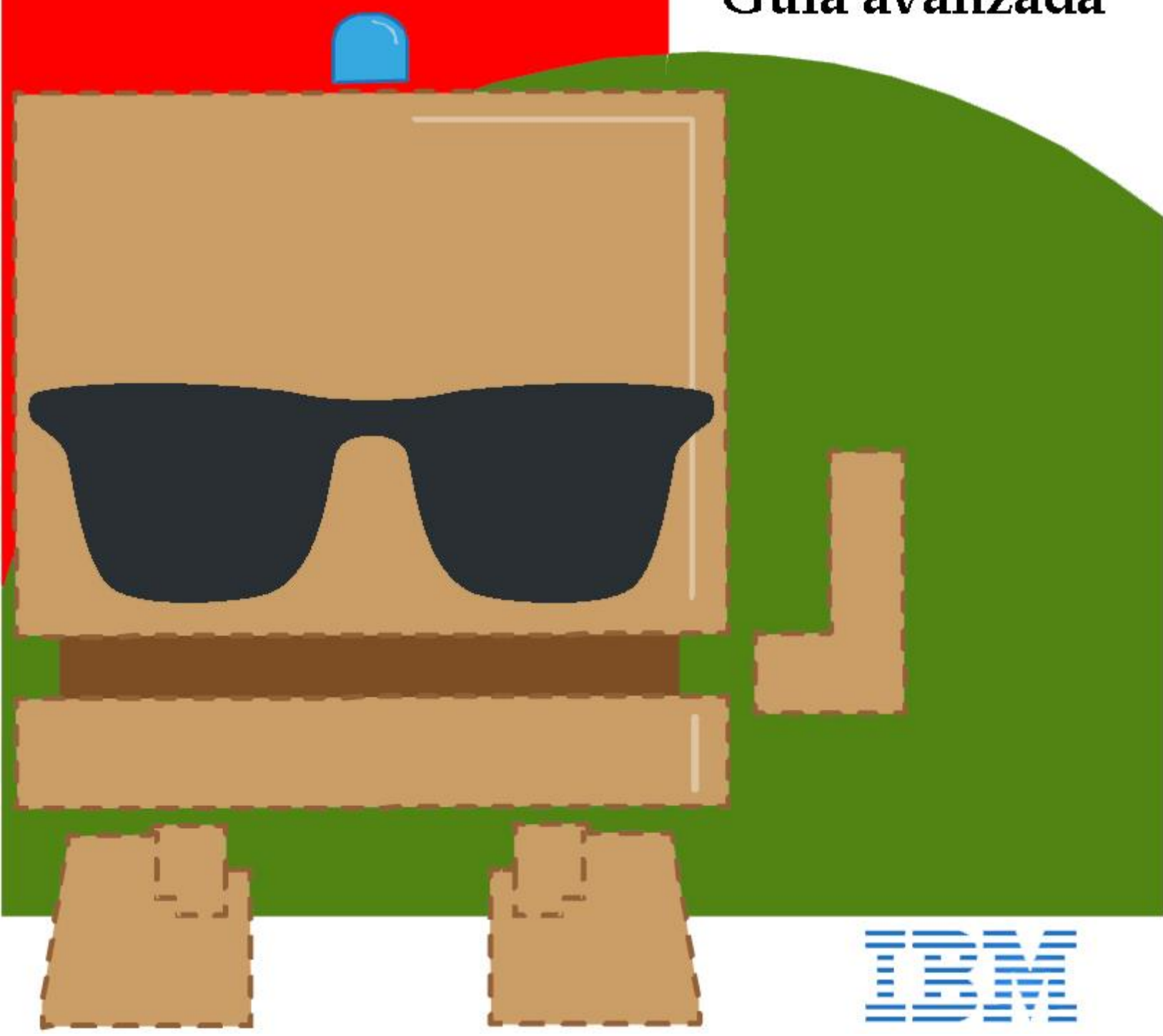


Explorando a TJBot:

Un vistazo al mundo de los robots

MANUAL PARA EL VOLUNTARIO DE IBM

Guía avanzada



IBM

Esta Unidad es parte de un Taller desarrollado en México.

Explorando a TJBot: Un vistazo al mundo de los robots

Manual de guía avanzada para el voluntario de IBM

La presente secuencia didáctica se realizó gracias a la iniciativa y apoyo financiero de IBM México.

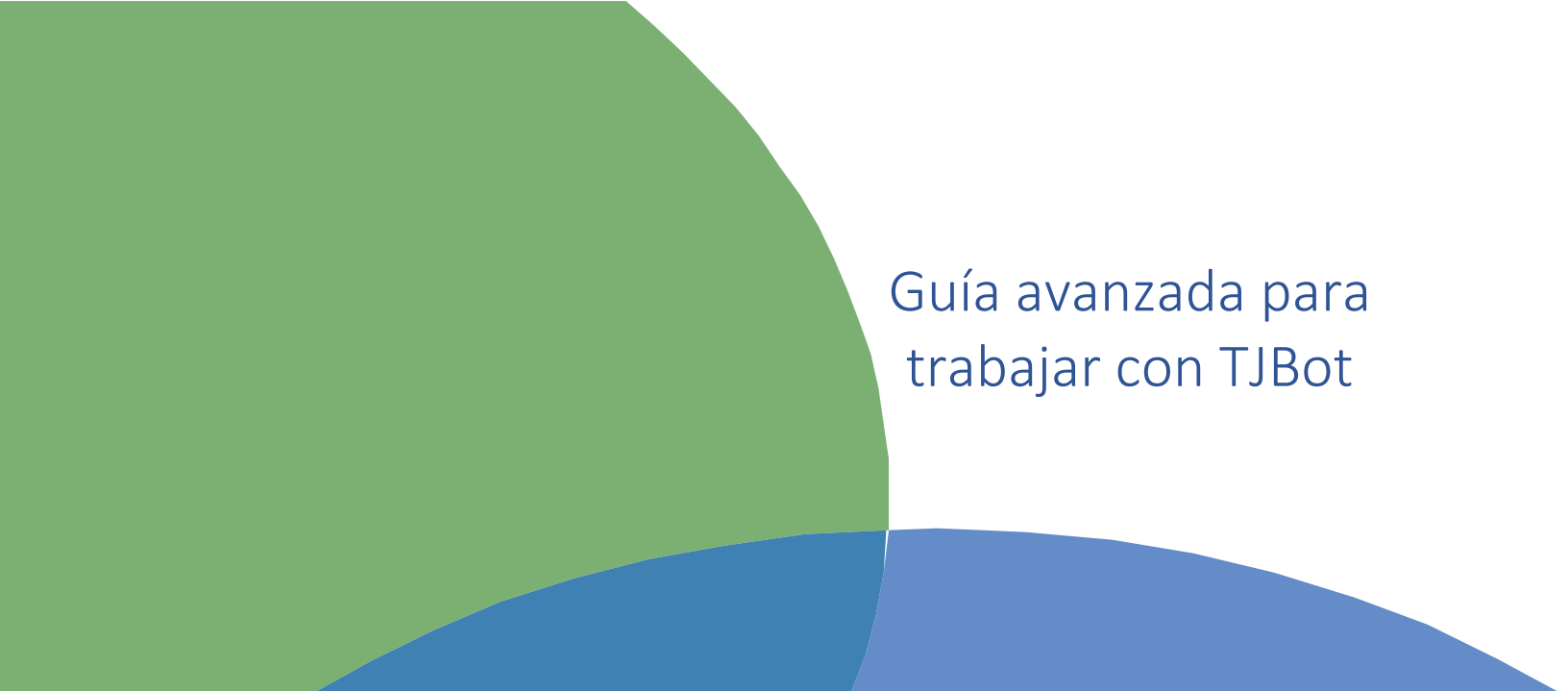
Esta secuencia didáctica fue desarrollada en el año 2018 por Jair Axel Lizarraga Velarde.



Índice

Contents

Introducción.....	5
Antecedentes	6
Requisitos previos, material de trabajo y actividades previas.....	11
Instrucciones para configurar a TJBot en Node-RED	13
Desarrollo de actividades.....	15
Instrucciones para que TJBot traduzca idiomas	16
Instrucciones para que TJBot hable lo que ve.	20
Instrucciones para que TJBot twittee el clima	22
Instrucciones para que TJBot reaccione a Twitter.....	29
Instrucciones para que TJBot twittee una selfie.....	32
Conclusiones.	34



Guía avanzada para trabajar con TJBot

Introducción

El manual básico nos demostró el correcto funcionamiento de TJBot para diversas pruebas haciendo uso de la interfaz gráfica Node-Red, y ahora conocemos algunas las habilidades de TJBot: Escuchar, mirar, encender su led, mover su brazo, y traducir; sin embargo, aún queda por explorar el potencial que Node-Red ofrece como una herramienta de trabajo para el Internet de las Cosas (IoT).

Node-Red contiene nodos que permiten la toma de decisiones de acuerdo con ciertos parámetros, conectividad con APIs en internet que ofrecen servicios de todo tipo, interacción con páginas web, y además la posibilidad de instalar nuevos nodos para expandir las capacidades de la tarjeta de desarrollo Raspberry Pi y de TJBot en sí.

Si bien Node-Red es una herramienta que trabaja sobre JavaScript para realizar sus funciones y así interactuar con los dispositivos, este sistema permite realizar programación sin necesidad de algún lenguaje de programación, lo que facilita su acercamiento a personas con muy pocos conocimientos técnicos.

En esta guía, descubriremos la manera de encadenar funcionalidades de las llamadas APIs y junto con TJBot, utilizando nodos conectados entre sí para lograr realizar un robot más interactivo, más funcional, y más divertido.



Antecedentes

Para extender las capacidades de nuestro robot, es conveniente entender algunos conceptos previos, que serán aplicados a lo largo de este taller.

¿Qué es un JSON?

El programa Node-Red hace uso del lenguaje de programación JavaScript, y para el intercambio de datos entre los flujos, nodos y el programa, se utiliza el formato de texto ligero llamado JSON.

JSON es texto plano que ofrece una estructura para el intercambio de información, su forma es del tipo llave-valor, donde el valor puede ser un valor simple u otro conjunto llave-valor. Esto permite consultar la información contenida conociendo únicamente la llave. Un JSON tiene la siguiente estructura:

1.- { llave1 : valor1 }

Además, puede contener llaves dentro de los valores:

2.- { llave1: { llave 2: valor 2} }

Para consultar el contenido del ejemplo 1, bastará con utilizar la siguiente notación:

msg.llave1

Para obtener el contenido de la llave 2 en el ejemplo 2, podremos escribir:

`msg.llave1.llave2`

¿Qué es un array?

Un array es una listado de elementos contenidos, en su representación, estos elementos se encuentran encerrados entre corchetes. Un ejemplo es:

`miLista = [elemento1, elemento2, elemento3]`

Para acceder a la información de un array, es necesario indicarlo de la siguiente manera:

`miLista.[0]`

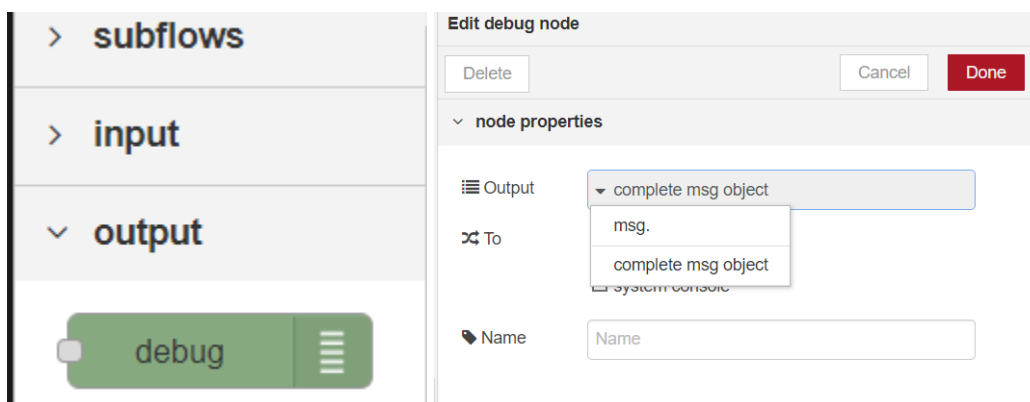
Donde el cero indica la posición inicial del array, un 1 indicaría la segunda posición, y así consecutivamente.

¿Cómo es la información en un nodo de Node-Red?

Como mencionamos anteriormente, los nodos reciben y transmiten información en formato JSON. El contenido de información varía de acuerdo con el nodo usado, pero generalmente contiene al menos estas dos llaves:

- **_msgid**: El identificador del mensaje.
- **Payload**: La carga que contiene el nodo en el momento.

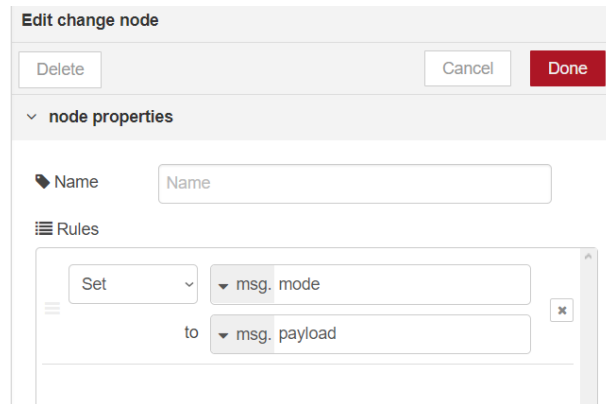
Una forma fácil de ver la información que emite un nodo es utilizar el nodo debug, el cual está contenido en el panel izquierdo en la sección output. Al dar doble clic en este nodo, podrás desplegar una lista llamada Output, con dos opciones. La primera, llamada "msg.", te permitirá ver la información de la llave que escojas, navegando entre las llaves y los arrays que el JSON contiene (Como lo viste en las explicaciones anteriores, para acceder entre llaves utilizarás el operador punto, mientras que, para acceder entre un array, utilizarás corchetes y la posición en el array de tu interés)



Un claro uso de esto, es en el uso del nodo change.

Nodo change:

El nodo change te permitirá intercambiar la información de un mensaje, cambiando el contenido de una llave por el de otra, reemplazar o incluso crear contenido nuevo.



En el ejemplo anterior, tenemos un menú desplegable en Set, que copiará el contenido del primer campo al segundo. Además, se cuenta con las opciones de Change (cambiar), Delete (eliminar), Move (mover), para manipular la información del JSON que cruce por el nodo.

Esto es útil, ya que los nodos de TJBot esperan que el mensaje a utilizar se encuentre en la llave *payload*, sin embargo el resultado de algunos nodos alojan el resultado en una llave distinta, y debe ser nuestra labor intercambiar esta llave por la llave *payload*.

Nodo switch:

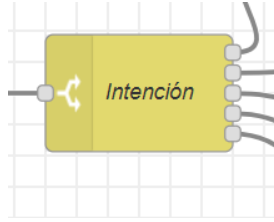
En principios de programación, se maneja un concepto llamado sentencias condicionales, y es el pilar de la programación estructurada. Es una instrucción que se puede o no ejecutar en función del valor de una condición.

Node-Red contiene un nodo llamado Switch. Dicho nodo aplica el concepto de sentencia condicional, de manera que de acuerdo con la entrada elige entre alguna de sus posibles salidas.

En el campo Property, se indica la entrada que el nodo comparará. Y en el menú desplegable, se indica el tipo de comparación a realizar. A continuación, se presentan las comparaciones más comúnmente utilizadas:

Símbolo	Significado
==	Igual a
!=	Diferente a
<	Menor a
>	Mayor a
<=	Menor o igual a
>=	Mayor o igual a
Is True	Es verdadero
Is False	Es falso
Is Null	El valor es vacío, o no fue asignado.
Otherwise	En cualquier otro caso

A un lado del menú desplegable, se encuentra el elemento con el que la entrada se comparará. Se pueden añadir tantas comparaciones como se desee, esto añadirá nuevas salidas del nodo.



Nodo con una entrada y cinco sentencias condicionales

¿Qué son las APIs?

Para propósitos de este manual, podemos pensar en la API como un software desarrollado por un tercero, que ofrece información o funcionalidades y está a disposición de las personas para obtener y enviar información a él.

Existen todos tipos de APIs, desde las que te permiten consultar las noticias, obtener música, información sobre el clima, entre otros. Para poder usar esos servicios necesitamos una llave de acceso, la cual puede ser un Token (Código de acceso), o un usuario y contraseña. Esto se puede obtener en la página de internet de la API que deseamos usar.

Por ejemplo, en el manual pasado hicimos uso de la API de Watson, para obtener servicios como reconocimiento de objetos, convertir la voz a texto, y traducir texto, donde para consumir esos servicios necesitábamos de un usuario y contraseña.

En realidad, TJBot se comunica con Watson a través de JSON, y lo puede hacer con cualquier otra API de esta manera de forma muy sencilla, como lo veremos a continuación.

Requisitos previos, material de trabajo y actividades previas.

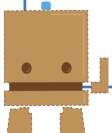
Requisitos previos:

Completar manual **Guía de instalación** de TJBot, para cada robot.

Consultar y opcionalmente realizar **Guía básica** de TJBot. (Para el armado del robot, y entender el uso de los nodos de Node-Red).

Material por TJBot (para cada equipo de 5 integrantes):

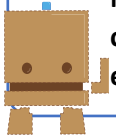
- 1 estructura de TJBot impresa en cartón
- Cinta adhesiva blanca traslúcida
- 1 pincel
- Pegamento blanco
- 1 mini micrófono USB.
- 1 computadora Raspberry Pi 3 Model B con cable.
- 3 cables jumper hembra/hembra de colores preferentemente negro, amarillo y verde.
- 3 cables jumper hembra/macho de colores preferentemente café, rojo y naranja.
- 1 Motor (micro servo).
- 1 tarjeta micro SD de 16GB de capacidad.
- 1 foco LED (neoPixel Diffused 8mm LED-x5).
- 1 cámara para Raspberry Pi NoIR Camera V2.



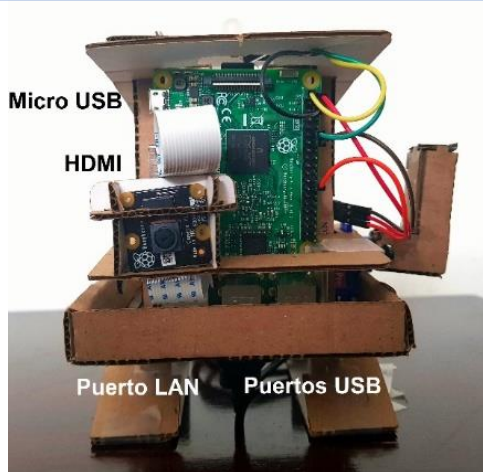
Nota: En caso de no disponer de una red Wifi, o que la configuración de red Wifi no se aplique correctamente, conectar un cable ethernet entre tu modem y TJBot para obtener internet.

Instrucciones para realizar la conexión con TJBot

1. Quiten el cartón que cubre a los componentes internos de **TJBot** (su cabeza).
2. Verifiquen que la cámara, micrófono, bocinas, motor y LED estén bien conectados a la computadora *Raspberry* de **TJBot** (ver hoja de instrucciones de armado).
3. Conectar la Raspberry pi a la corriente eléctrica.
4. Transcurridos aproximadamente 40 segundos, escucharás la dirección IP de TJBot. Anótala.
5. Con una computadora conectada a la misma red de internet que TJBot, abra el navegador web y escriba la dirección obtenida en el paso anterior, concatenando al final el texto **:1880**, por ejemplo, si la dirección que escuchaste es **192.168.1.10**, deberás anotar en el navegador web **192.168.1.10:1880**.
6. Deberá ver el espacio de trabajo de Node-red.



Nota: Es muy importante que lo último que se haga sea conectar a la corriente eléctrica ya que la computadora Raspberry no tiene botón de encendido como otras computadoras. La manera de encenderla es conectarla.



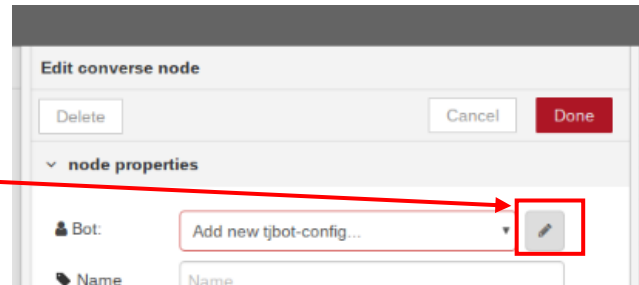
Conexiones de TJBot

Instrucciones para configurar a TJBot en Node-RED

1. Una vez en Node-red identifique en la barra lateral izquierda la sección “TJBot”. Elija cualquier nodo y arrástrelo al espacio de trabajo.

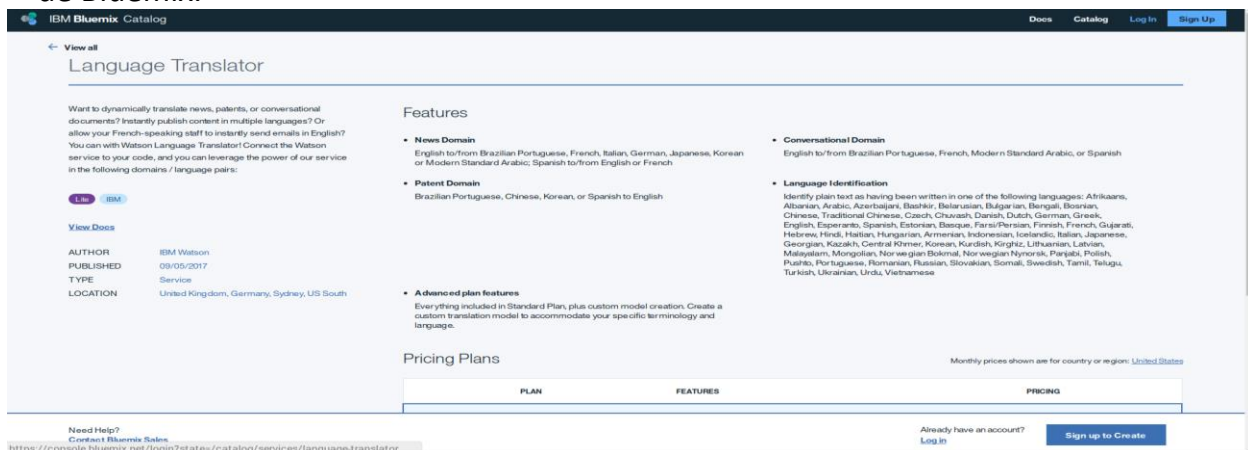
2. Dé doble clic al nodo

3. Dé clic al botón con un lápiz para para configurar a TJBot

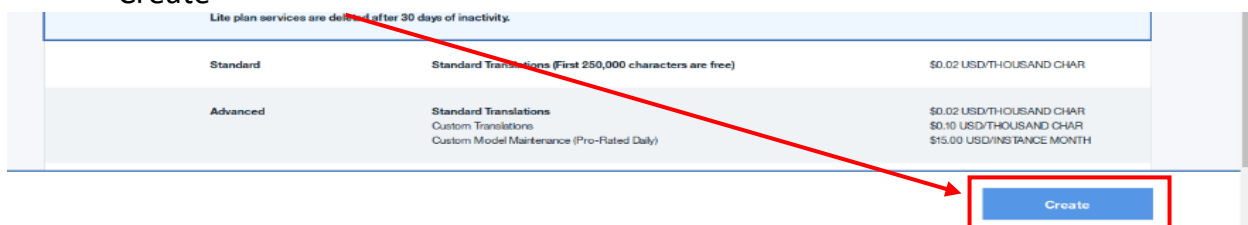


4. Seleccionen el género de TJBot, los idiomas que habla y escucha (Spanish North American) y seleccionen la casilla de todos los elementos que tiene (*Servo/motor, LED, Camera, Microphone, Speaker*). También, escriban TJBot en el espacio para el nombre, el nuestro lo llamaremos **María**.

5. En la parte inferior de esta misma ventana encontrarán los espacios para incluir las certificaciones de los servicios de Watson que TJBot puede utilizar. Den click al cuadrado con una flecha al lado de la sección “Language Translator” para abrir la página de Bluemix.



6. Para crear las credenciales diríjase a la parte inferior de la página y de click al botón “Create”



- En la página que se abrió, da click en el botón “show” para mostrar las credenciales. Selecciona el “username” y “password” y peguelos en la sección correspondiente del editor de Node-Red.

The image shows two screenshots from a tutorial. The top screenshot is from the Watson Language Translator-8z service interface. It displays the 'Credentials' section with a 'Show' button circled in red. Below it, a JSON snippet shows the 'username' and 'password' fields highlighted in yellow. The bottom screenshot is from the Node-RED 'Add new tjbtc-config config node' dialog. It shows various configuration options like Gender, Speak, Listen, and Tone Analyzer. The 'Language Translator' section at the bottom has its 'Username' and 'Password' fields highlighted with a red box. The 'Add' button is also circled in red.

- Repite los pasos del 5 al 8, para los servicios de Texto a habla (Text to Speech), Habla a texto (Speech to Text), Reconocimiento visual (Visual recognition), Traductor de lenguaje (Language translator), Análisis de sentimiento (Sentiment analysis)



Desarrollo de actividades

A continuación, se desarrollarán actividades más complejas de TJBot que harán uso de varios nodos a la vez, obtendremos información de APIs, manipularemos la información que pasa a través de los nodos, y tomaremos decisiones basados en su contenido.

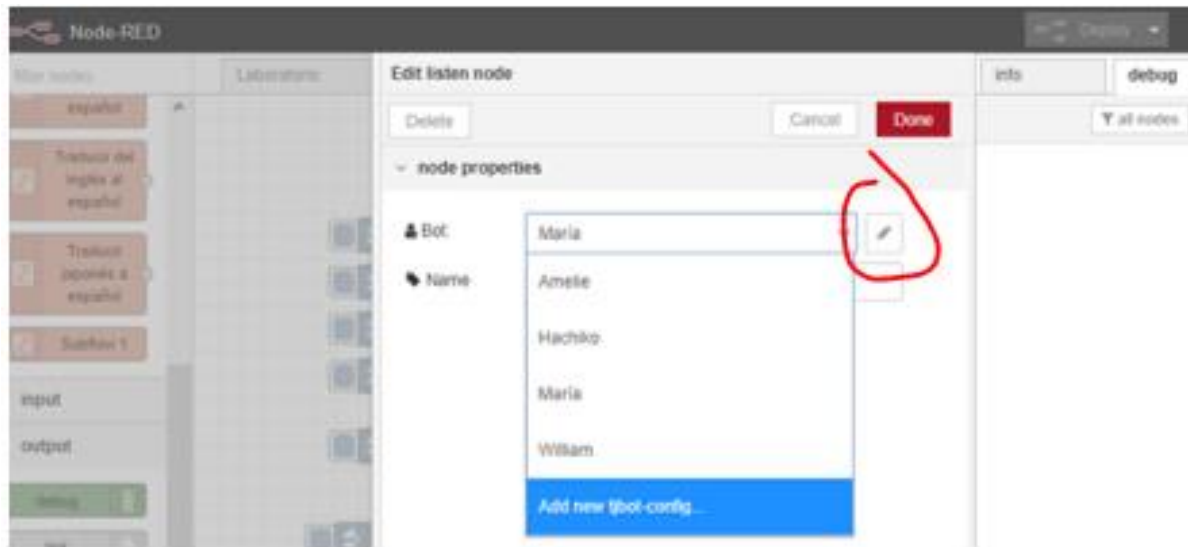
Entre las actividades que desarrollaremos, se encuentran:

- [Actividad: Hacer que TJBot traduzca idiomas.](#)
- [Actividad: Hacer que TJBot hable lo que ve.](#)
- [Actividad: Hacer que TJBot twittee el clima.](#)
- [Actividad: Hacer que TJBot reaccione a Twitter.](#)
- [Actividad: Hacer que TJBot twitee una selfie.](#)

Actividades:
Instrucciones para que TJBot traduzca idiomas

Instrucciones para que TJBot traduzca idiomas

Crearemos una nueva configuración de Bot. Sigue las instrucciones para configurar a TJBot en Node-Red para crear otro Bot más, seleccionando primeramente Add new TJBot config en la lista desplegable, y luego dando click al botón del lápiz para configurarlo.



Configure este nuevo Bot para escuchar y hablar en inglés. Agregue las credenciales **Speech to text** y **Text to speech**, use las mismas con las que creó a su Bot anterior, y en la sección **Name** escriba William.

Da click en Add, y finalmente en done.

Edit listen node > Edit tjb之-config node

Delete Cancel Update

Gender: Male

Speak: English (British dialect)

Listen: UK English

Has ☒ Servo ☒ LED ☒ Camera

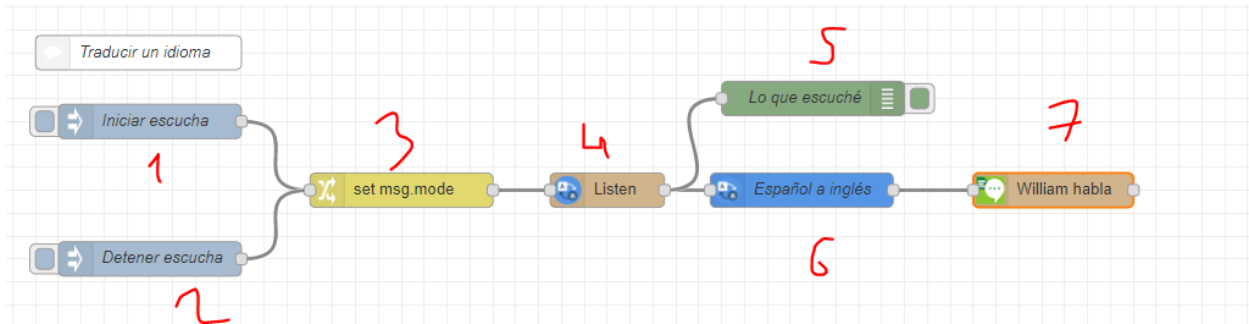
☒ Microphone ☒ Speaker

Name: William

A partir de este punto, tendremos dos Bots configurados: Al primero de ellos nos referiremos como María, y el segundo de ellos es William. María tiene configurado el lenguaje español, y es capaz de traducir idiomas. Mientras que William habla y escucha inglés, pero no es capaz de traducir.

Actividades:

Instrucciones para que TJBot traduzca idiomas

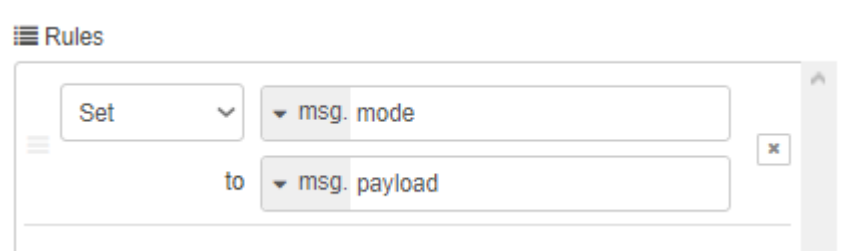


2.- Construya el siguiente flujo, arrastrando cada nodo requerido al espacio de trabajo, uniéndolo con el nodo correspondiente, y dando doble click a cada nodo para configurarlo de acuerdo con lo siguiente:

Nodo 1.- Del apartado input, nodo inject. **Payload:** string: start. **Name:** Iniciar escucha.

Nodo 2.- Del apartado input, nodo inject. **Payload:** string: stop. **Name:** Detener escucha.

Nodo 3.- Del apartado function, nodo change.



Configuración del nodo Change

Nodo 4.- Del apartado TJBot, nodo listen. **Bot:** María.

Nodo 5.- Del apartado output, nodo debug. **Output:** msg.payload. **Name:** Lo que escuché.

Este nodo nos permitirá leer en el panel de debug lo que TJBot ha escuchado.

Nodo 6.- Del apartado IBM Watson, nodo language translator. **Mode:** Translate. **Source:** Spanish. **Target:** English. **Name:** Español a inglés.

NOTA: Para el correcto funcionamiento de este nodo, agregaremos la API Key del servicio Lenguaje Translator obtenida de los servicios de Watson, en el campo **API Key** del nodo, esto habilitará el servicio de traducción.

Nodo 7.- Del apartado TJBot, nodo speak. **Bot:** William, **Mode:** speak, **Name:** William habla.

Actividades:

Instrucciones para que TJBot traduzca idiomas

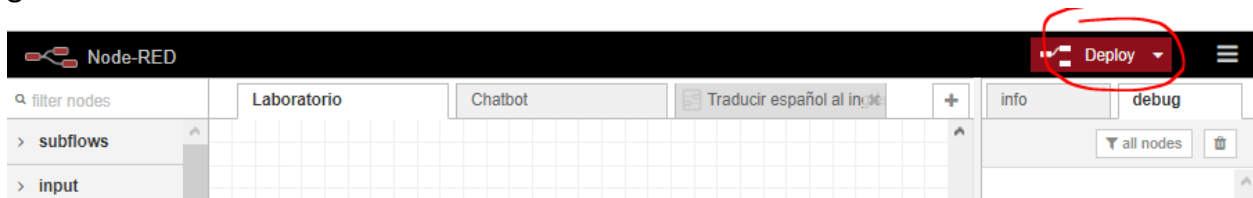
Descripción del flujo.

Los nodos **1, 2, 3 y 4** nos permitirán iniciar y detener el escucha del robot. Podemos ver que el nodo **3: *Change*** está cargando la instrucción ***stop y start*** que sale de la llave mode, a la llave payload, que es la que el nodo **4: *Listen*** es la que lee por default.

El nodo **5 *Lo que escuché*** nos permitirá ver en pantalla lo que nuestro robot escuchó por medio de su micrófono, el nodo **6: *Español a Inglés*** se encargará de realizar la traducción de lo que se ha escuchado el idioma objetivo, y finalmente el nodo **7: *William habla*** utilizará al bot configurado William para hablar en un acento inglés lo que se ha escuchado.

Puesta a prueba:

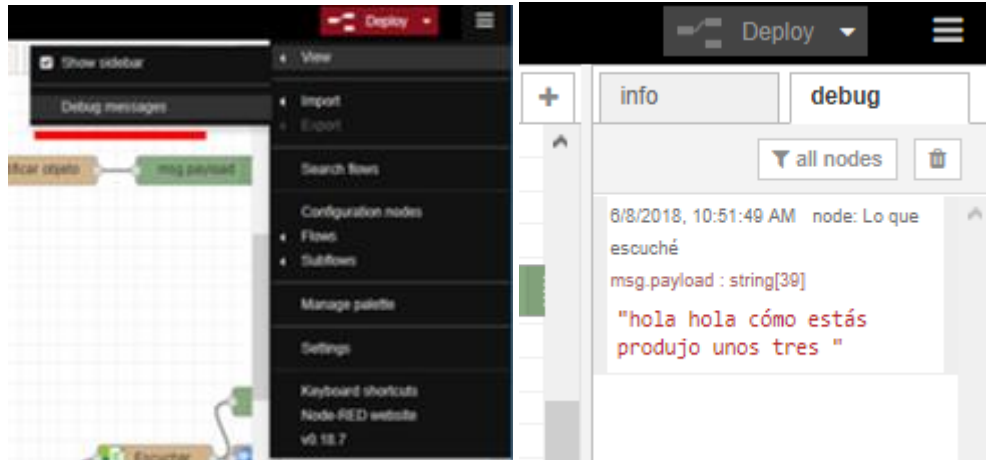
Ahora, da click al botón Deploy, que se encuentra en la esquina superior izquierda, esto guardará los cambios realizados.



Da click al botón azul del nodo ***Comenzar a escuchar***, acércate a TJBot y habla unos segundos con él. Escucharás como unos segundos después TJBot comienza a traducir al inglés lo que has dicho, o en caso de haber algún error, lo informará en español. Da click al botón ***Parar de escuchar*** cuando hayas acabado. Además, podrás corroborar si TJBot te escuchó correctamente abriendo la ventana Debug, dando click al menú desplegable de la esquina superior derecha, click en *view* y finalmente en *Debug messages*. Ahora consultando en el menú abierto del lado derecho, en la pestaña Debug:

Actividades:

Instrucciones para que TJBot traduzca idiomas



¿Qué más puedo hacer?

Ahora que puedes traducir del español al inglés, ¿Porqué no hacerlo del inglés al español?

- 1- En el nodo **Listen**, cambiar a María por William.
- 2- En el nodo **Español a inglés**, cambiar **Source** a **English**, y **Target** a **Spanish**, también cambia el nombre del nodo a **Inglés a español**.
- 3- En el nodo **William habla**, cambiar al Bot **William** por **María**, y cambiar el nombre del nodo a **María habla**.
- 4- Prueba una vez más dando click a **Deploy** y dando click al botón azul de **Comenzar a escuchar**. Habla en inglés alguna frase o palabra, y verás como María lo traduce al español. Cuando termines, da click al botón azul de **Parar de escuchar**.

Puedes explorar los diferentes idiomas que Watson tiene para traducir, como lo son italiano, portugués o francés. Para realizarlo, lleva a cabo las siguientes acciones:

- 1- En el nodo **Listen**, escoge a un Bot que esté configurado para el idioma en el que le hablarás a TJBot.
- 2- En el nodo de traducción, cambia el **Source** y el **Target** según corresponda.
- 3- En el nodo que habla lo que se ha traducido, escoge a un Bot configurado con el idioma de habla que deseas reproducir.

Hay algunas APIs que funcionan mejor cuando usas el idioma inglés para interactuar con ellas, como por ejemplo la funcionalidad de análisis de sentimientos. Puedes aprovechar la traducción de este flujo para interactuar con estas funciones.

Actividades:

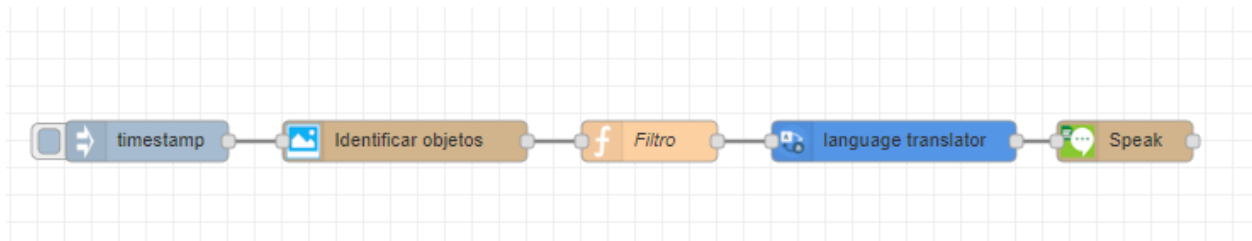
Instrucciones para que TJBot hable lo que ve.

Instrucciones para que TJBot hable lo que ve.

En esta actividad podremos poner a prueba las capacidades de reconocimiento visual con las que TJBot cuenta, para después manipular los datos obtenidos, de forma que podamos escuchar una descripción de como TJBot ve el mundo real a través de su peculiar ojo.

Vale la pena mencionar que los resultados obtenidos se encuentran en inglés, por lo que utilizaremos traducción para escuchar en español los resultados.

Construye el siguiente flujo, arrastrando el nodo solicitado al espacio de trabajo, uniéndolo al nodo correspondiente, y dando doble click para configurar cada nodo de acuerdo con las siguientes indicaciones:



Nodo 1.- Del apartado input, nodo inject. **Payload:** timestamp, **Name:** Ver objeto.

Nodo 2.- Del apartado TJBot, nodo see. **Mode:** See (identify objects), **Name:** Identificar objeto.

Nodo 3.- Del apartado function, nodo function. **Name:** Filtro.

Escribir en el espacio **Function** siguiente código:

```
var elementos = "En este momento estoy viendo...";

for(var elemento in msg.payload){
    elementos += msg.payload[elemento].class + ", ";
}
msg.payload = elementos;
return msg;
```

Este Código permitirá obtener uno a uno los elementos que TJBot observó, haciendo una cadena de texto que pueda después leer en voz alta.

Actividades:

Instrucciones para que TJBot hable lo que ve.

Descripción del flujo.

Este sencillo flujo inicia con un nodo **inject** que inicializa la acción de comenzar a observar. Cuando se acaba de analizar la imagen observada, los resultados se obtienen en forma de un JSON que contiene un array, cada array contiene un objeto y este a su vez contiene la información de lo que se ha visto. Para poder simplificar la extracción de este dato, creamos un nodo de función que, por medio de JavaScript, obtendrá cada elemento visto en la imagen, lo colocará y presentará en una cadena de texto que será traducida al español, pues recordemos que los resultados del reconocimiento visual llegan en inglés. Finalmente, TJBot dirá en voz alta los resultados obtenidos.

Puesta a prueba.

Da click al botón del nodo **Ver objeto** para iniciar a ver objetos. En ese momento, puedes colocar objetos frente a la cámara: Una botella, un lápiz, o tú mismo. Deberás esperar algunos segundos para la captura de la imagen, y una vez terminado, escucharás por el altavoz el resultado de todo lo que ha visto TJBot.

Actividades:

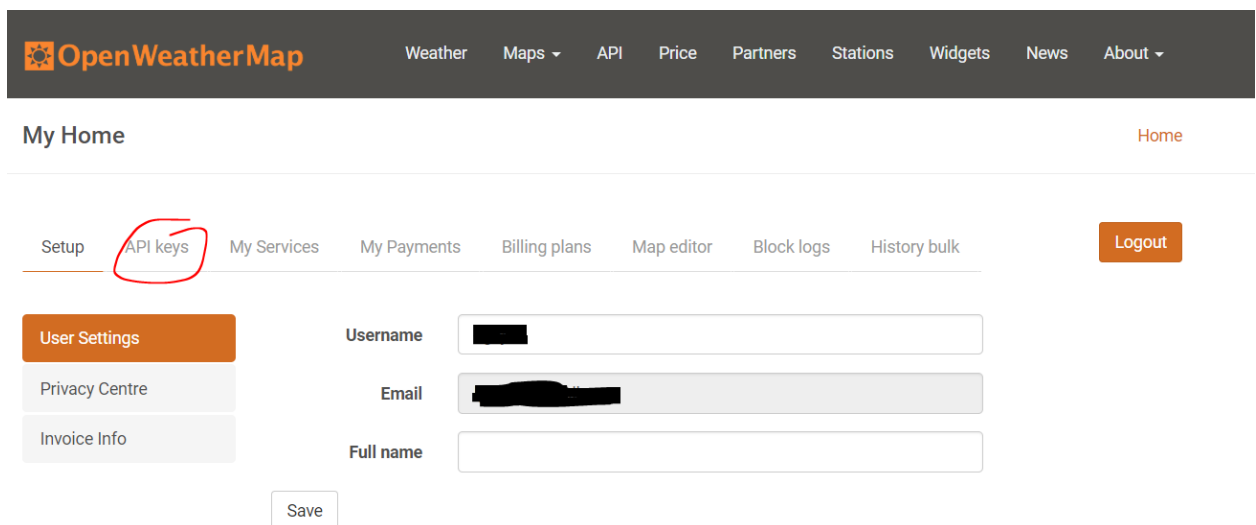
Instrucciones para que TJBOT twitee el clima

Instrucciones para que TJBOT twitee el clima

En esta actividad, TJBOT hará uso de la API de una página de información climática, openweathermap.org, para que sea informado en tu perfil de Twitter. Es requisito contar con una cuenta de twitter antes de proceder a la realización de esta actividad.

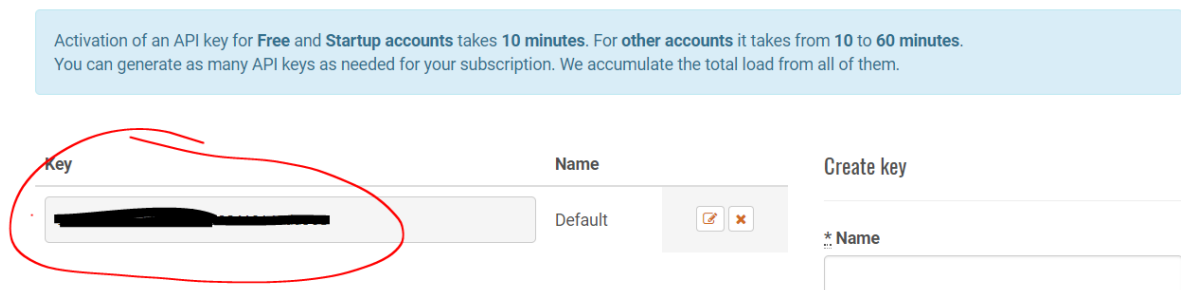
Primeramente, debemos obtener el permiso de acceso a su API. Para esto, realizaremos los siguientes pasos:

- 1- Crea una nueva cuenta en https://home.openweathermap.org/users/sign_up.
- 2- Automáticamente accederás a la página de información. Dirígete a **Api keys** de la misma pantalla:



The screenshot shows the OpenWeatherMap 'My Home' page. The top navigation bar includes the OpenWeatherMap logo and links for Weather, Maps, API, Price, Partners, Stations, Widgets, News, and About. The 'My Home' section has a 'Home' link. Below this, a horizontal menu contains 'Setup', 'API keys' (circled in red), 'My Services', 'My Payments', 'Billing plans', 'Map editor', 'Block logs', and 'History bulk'. A 'Logout' button is on the right. The 'User Settings' sidebar on the left includes 'User Settings' (selected), 'Privacy Centre', and 'Invoice Info'. The main content area shows fields for 'Username', 'Email', and 'Full name', with a 'Save' button below them.

- 3- Copia tu API key para utilizarlo más adelante.



The screenshot shows the 'Create key' form in the OpenWeatherMap API keys section. A blue notification box at the top states: 'Activation of an API key for Free and Startup accounts takes 10 minutes. For other accounts it takes from 10 to 60 minutes. You can generate as many API keys as needed for your subscription. We accumulate the total load from all of them.' Below this, the 'Key' field is circled in red. The 'Name' field is set to 'Default'. The 'Create key' button is visible. The 'Name' field is also labeled with a red asterisk and the text '* Name'.

Ahora llevaremos a cabo la instalación de los nodos del clima para Node-Red.

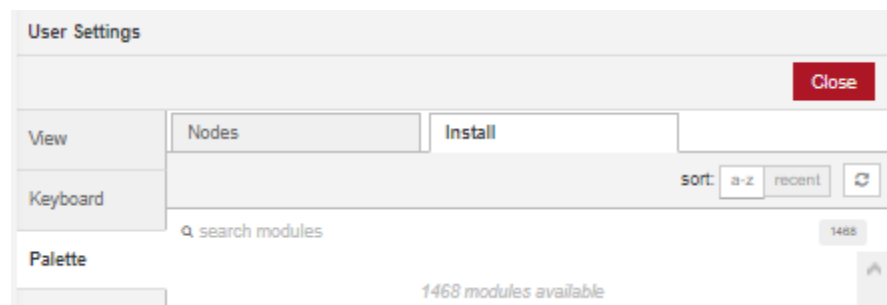
- 1- En la esquina superior izquierda, del menú desplegable selecciona **Manage palette**.

Actividades:

Instrucciones para que TJBot twitee el clima

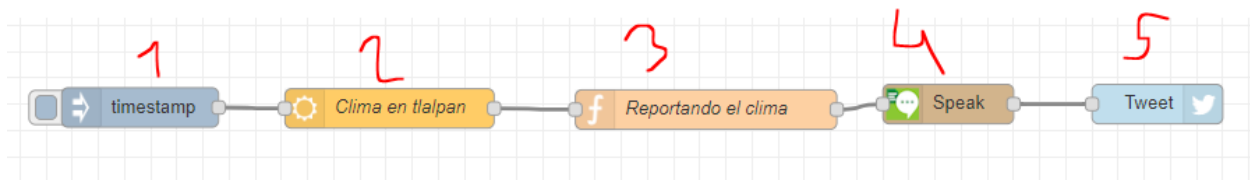


- 2- De las opciones a la izquierda del menú desplegado, selecciona **Palette**, y de las pestañas selecciona **Install**.



- 3- En el buscador **search modules** busca **node-red-node-openweathermap**, y da click en el botón Install para comenzar la instalación del nodo.
- 4- Se añadirán automáticamente un apartado de nodos llamado **weather**, con 2 nodos nuevos.

Finalmente, construiremos el flujo para llevar a cabo la actividad:



Nodo 1.- Del apartado input, nodo input. No será necesario configurarlo.

Nodo 2.- Del apartado weather, nodo openweathermap (con una entrada y una salida).

Actividades:

Instrucciones para que TJBot twitee el clima

API Key: La API Key que obtuvimos en la página de Open Weather.

El siguiente espacio permite escoger entre el clima al momento o lo pronosticado hasta dentro de cinco días. Escogeremos al momento.

Location: Permite escoger por ciudad o por coordenada, escogeremos por ciudad y colocaremos la ciudad de nuestro interés.

Nodo 3.- Del apartado function, el nodo function. Por nombre, escribiremos **Reportando el clima**, y en función, copiaremos y pegaremos el siguiente código:

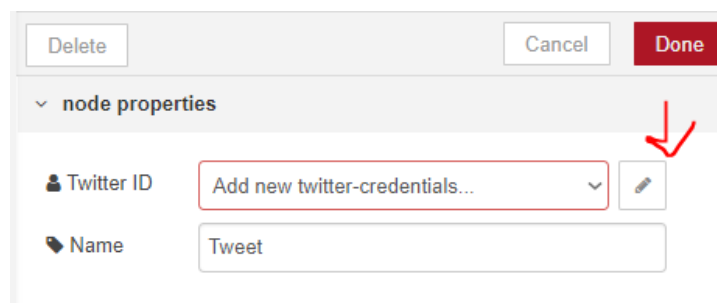
```
msg.payload = "El reporte del clima es, " + msg.payload.detail
+ ", tenemos una temperatura de " + msg.payload.tempc
+ "grados, y humedad máxima del " + msg.payload.humidity
+ "por ciento.";
return msg;
```

Este código construye texto basado en los resultados del nodo anterior. Como podemos ver, se accede a diversas llaves del JSON para obtener la información de nuestro interés.

Nodo 4.- Del apartado TJBot, nodo speak. **Bot:** María, **Mode:** Speak, **Name:** Reportar.

Nodo 5.- Del apartado social, nodo twitter con una entrada.

Para configurar este nodo, daremos doble click en él, y en las propiedades del nodo, click en el lápiz a un costado de “Add new twitter-credentials”.



A continuación, nos desplegará los campos a completar.

Actividades:

Instrucciones para que TJBot twitee el clima

Edit twitter out node > Add new twitter-credentials config node

Twitter ID @

1. Create your own application at apps.twitter.com

2. From the 'Keys and Access Tokens' section, copy the Consumer Key and Secret

Consumer Key

Consumer Secret

3. Create a new 'Access Token' and copy the Access Token and Secret

Access Token

Access Token Secret

Damos click al link que se encuentra en esta vista: ***apps.twitter.com*** y asegúrate de haber iniciado sesión. Aparecerá la siguiente vista, daremos click en el botón “Create new app”.

 Application Management



Twitter Apps

You don't currently have any Twitter Apps.

Llenaremos los campos que nos pide twitter como apunta a continuación:

En **Name**, escribe cualquier nombre que gustes, si la verificación falla, prueba con otro, ya que puede estar ocupada por otra aplicación de Twitter. Trata de escribir un nombre único, utilizando números para que sea difícil que alguien más lo repita.

Actividades:

Instrucciones para que TJBot twitee el clima

Create an application

Application Details

Name *

Your application name. This is used to attribute the source of a tweet and in user-facing authorization screens. 32 characters max.

Description *

Your application description, which will be shown in user-facing authorization screens. Between 10 and 200 characters max.

Website *

Your application's publicly accessible home page, where users can go to download, make use of, or find out more information about your application. This fully-qualified URL is used in the source attribution for tweets created by your application and will be shown in user-facing authorization screens.
(If you don't have a URL yet, just put a placeholder here but remember to change it later.)

Callback URLs

Where should we return after successfully authenticating? OAuth 1.0a applications must explicitly specify their oauth_callback URL(s) here, as well as include the one of the URLs below in the request token step. To restrict your application from using callbacks, leave this field blank.



[Add a Callback URL](#)

Developer Agreement

☒ Yes, I have read and agree to the [Twitter Developer Agreement](#).

[Create your Twitter application](#)

Asegurate de marcar el campo “Developer agreement”, y da click en “Create your twitter application. Una vez creada la aplicación, dirígete a la pestaña “Keys and Access Tokens”.

 Application Management 

Aplicación del clima

[Test OAuth](#)

[Details](#) [Settings](#) [Keys and Access Tokens](#) [Permissions](#)

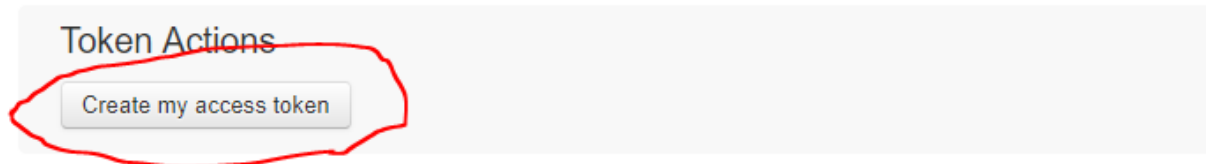
En la parte inferior de la pantalla verás la sección “Your Access Token”, ahí da click al botón “Create my Access token”

Actividades:
Instrucciones para que TJBot twitee el clima

Your Access Token

You haven't authorized this application for your own account yet.

By creating your access token here, you will have everything you need to make API calls right away. The access token generated will be assigned your application's current permission level.



Entonces se habrán creado automáticamente los tokens de acceso. En esta ventana, encontrarás las credenciales necesarias para nuestro nodo de twitter:

Aplicación del clima

Details Settings Keys and Access Tokens Permissions

Application Settings

Keep the "Consumer Secret" a secret. This key should never be human-readable in your application.

Consumer Key (API Key)

Consumer Secret (API Secret)

Access Level

Owner

Owner ID

Read and write (modify app permissions)

Jair_LizarragaV

142041689

Application Actions

Regenerate Consumer Key and Secret Change App Permissions

Edit twitter out node > Add new twitter-credentials config node

Cancel

Twitter ID

@

1. Create your own application at apps.twitter.com

2. From the 'Keys and Access Tokens' section, copy the Consumer Key and Secret

3. Create a new 'Access Token' and copy the Access Token and Secret

Consumer Key

Consumer Secret

Access Token

Access Token Secret

No olvides agregar tu TwitterID. Una vez llenados los campos del nodo, da click en Add, Done y Deploy.

Actividades:

Instrucciones para que TJBot twitee el clima

Descripción del flujo:

El primer nodo dará el banderazo para iniciar el proceso, y activará el nodo del clima. Este, por medio de la API key, accederá a la información de la API de Open Weather, y le solicitará información de la ciudad que le indicamos. Estos datos están en formato JSON, así que el tercer nodo se encarga de construir por medio de JavaScript el texto que queremos que sea dicho, a su vez accedemos al JSON para incrustar información en nuestro texto. El cuarto nodo dirá en voz alta el resultado, y por último el quinto nodo realizará una publicación en tu perfil de twitter de lo que TJBot acaba de decir en voz alta.

Puesta a prueba:

Bastará con dar click al botón azul del nodo input, para que se realice la acción. Espera unos segundos, y escucharás el reporte del clima por la bocina. Cuando termine el flujo, visita tu perfil de twitter, y verás que automáticamente se ha realizado una publicación en tu perfil acerca del clima.

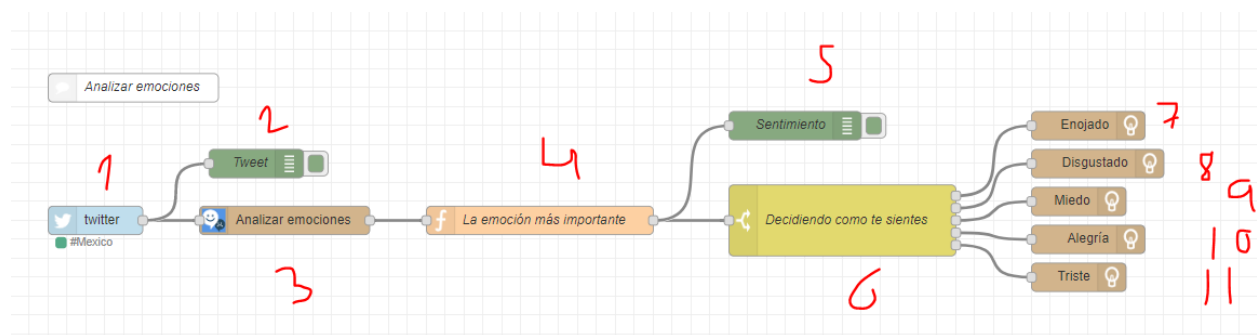
Actividades:

Instrucciones para que TJBot reacciona a Twitter

Instrucciones para que TJBot reaccione a Twitter.

A continuación, descubriremos la funcionalidad de TJBot que le permitirá identificar los sentimientos de lo que se le publica en tiempo real en twitter, y reaccionar a ellos por medio de su Led. Cuando usamos el nodo *analyze tone*, podemos analizar emociones, el lenguaje y el aspecto social del comportamiento. En esta actividad analizaremos las emociones, las cuales son clasificadas en cinco: Enojo, angustia, miedo, alegría y tristeza, cada una tiene una puntuación del cero al uno, de cuánto de ese sentimiento se detecta en la persona que está hablando. Para proceder con esta actividad, necesitarás una cuenta de twitter y haber [configurado exitosamente un nodo de twitter](#).

Construye el siguiente flujo, arrastrando el nodo solicitado al espacio de trabajo, uniéndolo al nodo correspondiente, y dando doble click para configurar cada nodo de acuerdo con las siguientes indicaciones:



Nodo 1.- Del apartado social, nodo Twitter. **Twitter ID:** Tu twitter id, **Search:** all public tweets, **For:** #Mexico. En el espacio For podrás poner la palabra que gustes. Utiliza la cuenta de twitter que configuraste en las **Instrucciones para que TJBot twitee el clima**.

Nodo 2.- Del apartado output, nodo Debug. **Name:** Tweet.

Nodo 3.- Del apartado de TJBot, nodo analyze tone. **Bot:** María, **Tones:** Emotion, **Name:** Analizar emociones.

Nodo 4.- Del apartado function, nodo function. **Name:** La emoción más importante. Escribir en el espacio **Function** la siguiente función:

```
msg.payload = msg.response.tones.reduce( function(a, b) {return (a.score > b.score) ? a : b;
}).tone_name;
return msg;
```

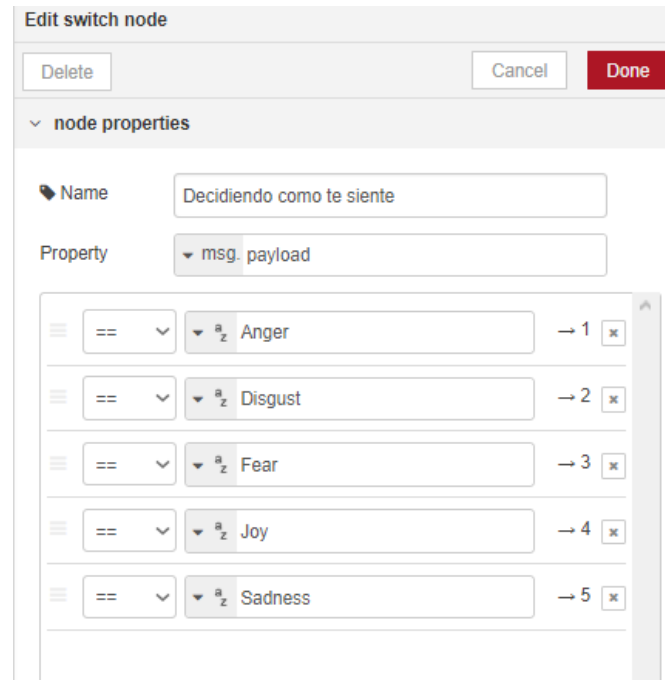
Este Código se encarga de seleccionar la emoción con más puntuación del análisis obtenido.

Actividades:

Instrucciones para que TJBot reacciona a Twitter

Nodo 5.- Del apartado output, nodo Debug. **Name:** Sentimiento.

Nodo 6.- Del apartado function, nodo switch. Se analizarán cinco opciones, una para cada sentimiento. La configuración es la siguiente:



Edit switch node

Delete Cancel Done

node properties

Name Decidiendo como te sienta

Property msg.payload

==	a_z Anger	→ 1	x
==	a_z Disgust	→ 2	x
==	a_z Fear	→ 3	x
==	a_z Joy	→ 4	x
==	a_z Sadness	→ 5	x

Nodo 7.- Del apartado TJBot, nodo shine. Mode: Shine, Color: red, Name: Enojado.

Nodo 8.- Del apartado TJBot, nodo shine. Mode: Shine, Color: green, Name: Disgustado.

Nodo 9.- Del apartado TJBot, nodo shine. Mode: Shine, Color: magenta, Name: Miedo.

Nodo 10.- Del apartado TJBot, nodo shine. Mode: Shine, Color: yellow, Name: Alegría.

Nodo 11.- Del apartado TJBot, nodo shine. Mode: Shine, Color: blue, Name: Triste.

Descripción del flujo.

El nodo de Twitter ha sido configurado para traer tweets relacionados con México, por medio del hashtag #Mexico, por lo que cada que se realiza una publicación en lared con este hashtag se detectará casi en tiempo real en nuestro flujo, y la información contenida en el twitt se enviará al analizador de emociones, procesando el texto e identificando las cinco emociones a analizar. Ahora es necesario escoger la emoción más relevante, por lo que la función (escrita en JavaScript) se encarga de seleccionar la emoción con el valor más alto. Finalmente tenemos un nodo switch, el cual comprobará cuál de las cinco emociones fue la seleccionada, y de acuerdo con la emoción, activará el

Actividades:

Instrucciones para que TJBot reacciona a Twitter

flujo correspondiente y encenderá el Led con alguno de los cinco colores que los representa.

Puesta a prueba.

Al dar click en Deploy para guardar los cambios, el nodo de Twitter comenzará a funcionar automáticamente trayendo los tweets que coincidan con la palabra introducida. En la vista de debug, podremos ver los tweets que se están recibiendo, y la reacción que TJBot tiene al respecto.

Cabe considerar que el campo de Twitter de búsqueda permite la búsqueda por palabras, hashtags y usuarios, y si la etiqueta de búsqueda trae muchos resultados, es posible que TJBot comience a mostrar lentitud en sus tareas, por lo que para detener la tarea es conveniente eliminar las conexiones del nodo de twitter y dar click en Deploy inmediatamente, ya que el nodo Twitter no tiene forma de ser parado por medio de alguna señal.

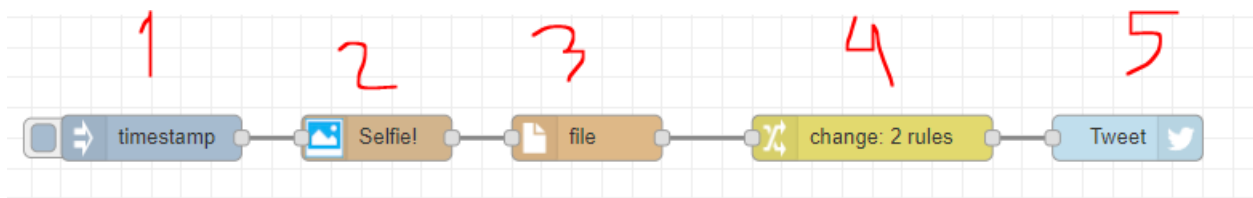
Actividades:

Instrucciones para que TJBot twittee una selfie.

Instrucciones para que TJBot twittee una selfie.

En esta actividad construiremos un flujo que utilizará la cámara de TJBot para tomar una foto y subirla a nuestra red social. Para proceder con esta actividad, necesitarás una cuenta de twitter y haber [configurado exitosamente un nodo de twitter](#).

Construye el siguiente flujo, arrastrando el nodo solicitado al espacio de trabajo, uniéndolo al nodo correspondiente, y dando doble click para configurar cada nodo de acuerdo con las siguientes indicaciones:



Nodo 1.- Del apartado Input nodo inject.

Nodo 2.- Del apartado TJBot, nodo See. **Bot:** maría, **Mode:** Take photo. **Flip:** Horizontal. **Dimensions:** 960x720, **Name:** Selfie.

Nodo 3.- Del apartado social, nodo file. **Output:** a single buffer object.

Nodo 4.- Del apartado function, nodo change. Configuración:

La imagen muestra la interfaz de configuración para el nodo 'change'. El título es 'Edit change node'. Hay botones para 'Delete', 'Cancel' y 'Done'. Debajo, se expande la sección 'node properties'.

En la sección 'Name', hay un campo de texto con el valor 'Name'.

En la sección 'Rules', hay una lista de reglas configuradas:

- Regla 1: 'Set' (mensaje) a 'msg. media' y 'to' 'msg. payload'.
- Regla 2: 'Set' (mensaje) a 'msg. payload' y 'to' 'Hola desde #TJBot'.

Actividades:

Instrucciones para que TJBot twittee una selfie.

Nodo 5.- Del apartado social, nodo Twitter de una entrada. ***Twitter ID:*** El perfil de twitter que configuraste en la actividad pasada. Utiliza la cuenta de twitter que configuraste en las **Instrucciones para que TJBot twitee el clima.**

Descripción del flujo: El nodo 1 se encargará de dar la orden para comenzar con el flujo. El nodo 2 está configurado en modo foto, la información de esta foto fluirá en modo buffer y será necesario almacenar estos datos en un archivo temporal, el cual es generado por el nodo 3, el cual recibe los datos a modo de buffer para generar la imagen que hemos tomado, el nodo 4 se encarga de pasar la imagen a la llave payload y de añadir algo de texto a payload para la publicación. Finalmente, el nodo 5 publicará en twitter la foto obtenida.

Puesta a prueba:

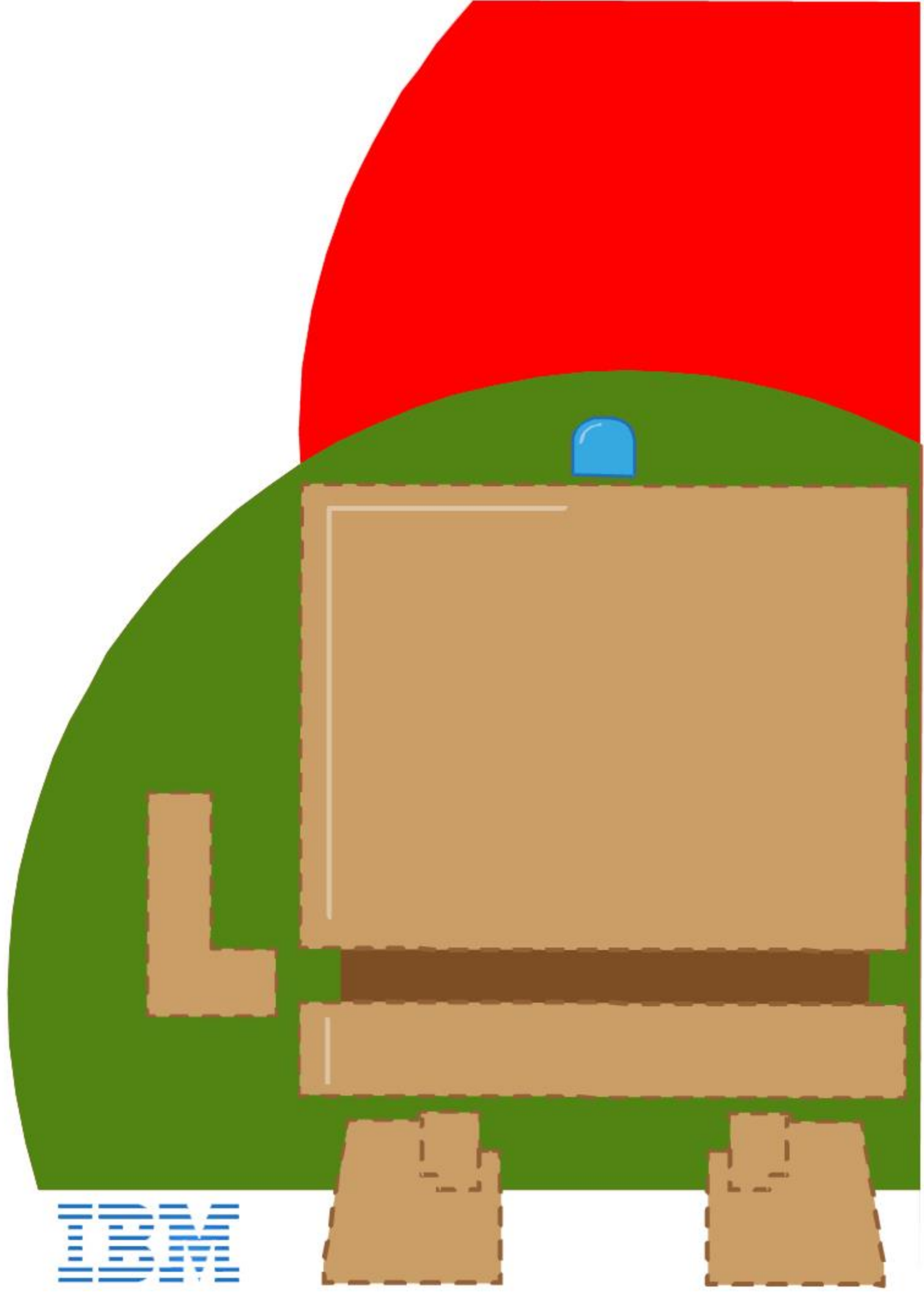
Bastará con dar la señal de inicio en el primer nodo para que el flujo se lleve a cabo, has tu mejor pose frente a la cámara y espera a que el flujo termine de ejecutarse para ver los resultados en tu perfil de twitter.

Conclusiones.

Como hemos podido comprobar con las actividades anteriores, TJBOT tiene una amplia gama de funciones que permiten una interacción entre el mundo exterior, el internet y sus componentes internos, los cuales alcanzan su máximo potencial combinando poco a poco los nodos de los que disponemos para que lleven tareas específicas y ordenadas.

Podemos apreciar que las combinaciones de diversos flujos pueden permitirnos llevar a cabo tareas más complejas de manera más cómoda, y basta realizar la conexión correcta entre los nodos. Por ejemplo, hemos realizado la conexión de los flujos donde TJBOT nos escucha, para enviar esta información al flujo que nos permite traducir idiomas. Por otro lado, hemos realizado un flujo que te platica lo que está viendo, pero como la API obtiene estos datos en inglés, lo hemos conectado a la traducción del inglés al español. ¡Existen toda clase de conexiones que harán a TJBOT un robot aún más potente!

Es aquí cuando vale la pena mencionar que una de las funcionalidades que TJBOT tiene es el de implementar un Chatbot por medio del nodo converse. La implementación de un Chatbot en TJBOT nos permitirá establecer conversaciones casi humanas con TJBOT, ordenar que realice actividades, traducciones o interaccione en internet, como si se tratara de una persona, la cual te entiende y realiza aquello para lo que le programes. Combinando las funcionalidades de TJBOT con los flujos que acabas de construir y un Chatbot, podrás realizar un asistente personal para que lleve a cabo lo que tu quieras.



IBM