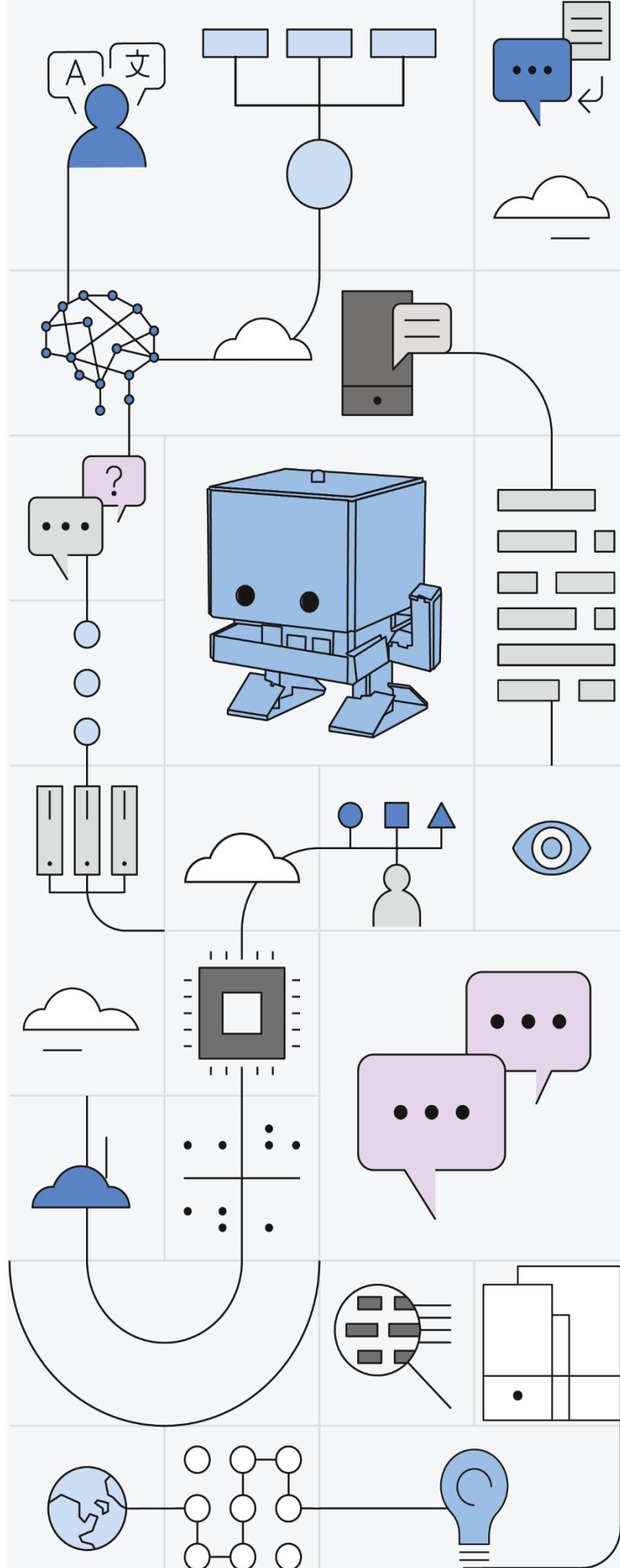


¡Hola TJBot!

Manual de usuario



Índice

Introducción	4
Antecedentes	5
¿Qué es Watson de IBM?	6
¿Qué es el cómputo cognitivo y cómo se relaciona con Watson?	7
Servicios de la nube	8
¿Qué es una API?	8
¿Qué es un proyecto de código abierto?	9
¿Qué es una Raspberry PI?	9
¿Cómo se trabaja con Raspberry PI?	10
¿Qué es Node-RED, y en qué lenguaje de programación está basado?	11
¿Qué es el formato de intercambio de información JSON?	11
TJBot	12
Preparación del Software	14
Paso 1: Instalando el sistema operativo Raspbian.	14
Paso 2: Instalando el proyecto TJBot	18
Conexión remota con TJBot: Node-RED	20
Aregar redes WiFi a TJBot	22
Instrucciones para usar el teclado en pantalla	24
Instalación rápida de TJBot con copia de seguridad	25
Formatear o reinstalar un TJBot ya grabado	26
Construyendo a TJBot	28
Objetivos	28
Material necesario por TJBot	28
Construcción de la carcasa de TJBot	30
Programando con TJBot	34
Conexión remota a TJBot con Node-RED	34
Instrucciones para crear cuenta en IBM Cloud	37
Configurando a TBot en Node-RED	38
Mi primer programa	40
TJBot, mueve tu brazo	45

TJBot, enciende tu led	48
TJBot... ¡Habla!	50
TJBot... ¡Escucha!	51
TJBot... ¡Observa!	55
TJBot... ¡Traduce!	57
Actividades avanzadas 59	
Nodo change	60
Nodo switch	61
Instalación de nuevos nodos	62
Instrucciones para que TJBot escuche y traduzca entre idiomas	63
Instrucciones para que TJBot describa lo que ve	68
Instrucciones para que TJBot twittee el clima	69
Instrucciones para que TJBot reaccione a Twitter	76
Instrucciones para que TJBot twittee una selfie	79
Anexos y Troubleshooting 80	
Conexión a TJBot por VNC (Escritorio remoto)	80
Conexión a TJBot por SSH (Ventana de comandos remota)	82
La ventana de comandos como fuente de resolución de problemas de Node-RED	84
TJBot no habla	87
Eliminando programación defectuosa de Node-RED por medio de la ventana de comandos	89
Configurar el volumen de TJBot	90
Configurar el volumen de TJBot cuando dicta la dirección IP	92
Eliminar el dictado de la dirección IP	93
Modificar el intervalo de comprobación de estatus de red en TJBot	94
No funciona la cámara de TJBot: Troubleshooting	94
Programación para ir más allá	96
Cupones para profesores y estudiantes en IBM Cloud	98
Glosario	99

Introducción

Cuando los estudiantes de cualquier nivel educativo construyen un robot están inmersos en un mundo multidisciplinario, donde la geometría, la trigonometría, la electrónica, la programación, el control, la mecánica, etc., proporcionan las capacidades básicas que redundarán en el éxito de esta actividad integradora conocida como práctica STEM (prácticas de ciencia, tecnología, ingeniería y matemáticas). Aparte de los conceptos que les son propios, la robótica da la oportunidad de desarrollar áreas como son: la ingeniería de sistemas, el diseño, conceptos de ergonomía del trabajo y la planificación.

En este sentido, IBM desarrolló un robot llamado TJBot como un proyecto de código abierto DIY (Do it on your own: Hazlo tú mismo) para construir cosas de manera divertida y fácil empleando la plataforma de servicios de Inteligencia Artificial Watson de IBM. TJBot fue nombrado así en honor al sucesor del primer presidente y Director Ejecutivo de IBM. Fue creado por Maryam Ashoori en IBM Research y presentado el 9 de noviembre de 2016.

TJBot al ser un proyecto de código abierto está al alcance de todo aquel interesado en practicar y conocer más acerca de la robótica y temas relacionados. En el presente manual enfocado a tutores, profesores y voluntarios, TJBot se utiliza para apoyar el proceso de enseñanza-aprendizaje en el tema de la robótica para estudiantes de 11 años en adelante.

A lo largo del desarrollo del manual, el tutor obtendrá los conocimientos necesarios para construir a TJBot, instalar su software, conectarlo a los servicios de la nube y programar con las actividades de acuerdo con la audiencia para la que se prepare, ya que al final del manual las actividades aumentan progresivamente de complejidad para demostrar cuán lejos puede llegar este robot.

La robótica es una materia en la actualidad en auge, está permeando en las instituciones educativas de formas muy diversas, se han incluido en los planes educativos para explicar cómo funciona la programación, cómo funcionan las computadoras, razonamiento lógico y matemático, todo esto como una manera innovadora de introducir a los estudiantes al futuro que nos avecina.

Dudas, comentarios y sugerencias:

alizarra@mx1.ibm.com

Para más información:

<https://cognitiveparatodos.mybluemix.net>

Antecedentes

El estudio de TJBot abre las puertas a temas que son nuevos o poco explorados por escuelas hoy en día, y es que la facilidad con la que este proyecto permite interactuar con distintas tecnologías obliga a tener un panorama abierto y claro sobre lo que se está por aprender. En este apartado abordaremos brevemente el contexto general necesario para abordar con suma seguridad el proyecto TJBot.

Un robot funciona formando un enlace entre los aspectos programables de su computadora (cerebro) y los aspectos interactivos con el entorno (cuerpo). Con una diversidad tan amplia de robots, pueden ser tan complejos de explicar como lo es explicar cómo funciona el organismo de un animal. De manera general podemos decir que un robot tiene dispositivos de entrada que le permiten recibir información de su entorno (por ejemplo, cámaras y micrófonos), dispositivos de salida que le permiten ejecutar acciones como producir sonidos, emitir luz y moverse (bocinas, luces LED y motores) y una computadora que decide las acciones a realizar dada la situación. Integrar todos estos componentes en el cuerpo de TJBot, reforzará la comprensión de las características de un robot e incentivará el trabajo colaborativo.

Para que los robots puedan realizar alguna tarea necesitan ser programados. La programación de un robot es un proceso crucial para que el robot responda correctamente a las instrucciones que se le brinden. El sistema operativo que está instalado en la tarjeta **Raspberry Pi** es llamado **Raspbian**. Un sistema operativo es un conjunto de programas informáticos que permiten la administración eficaz de los recursos de una computadora a través de la ejecución de aplicaciones con propósitos específicos. **Raspbian** está basado en Debian y es una distribución del sistema operativo GNU/Linux, por lo tanto, es libre y de código abierto.

El programa o herramienta de software que utilizaremos para programar a **TJBot** se denomina **Node-RED**. Node-RED fue desarrollada por el equipo de **Tecnologías Emergentes de IBM (IBM Emerging Technology)** para conectar dispositivos hardware, APIs y servicios en línea. Utiliza un editor basada en navegador WEB en el que de manera sencilla se pueden conectar **nodos (nodes)** para formar **flujos (flows)** de eventos. Los **nodos** contienen instrucciones ya programadas para facilitar la interacción de dispositivos, que conectados a internet forman el **Internet de las Cosas** (IoT por sus siglas en inglés). Gracias a las aportaciones de la comunidad de desarrolladores, el catálogo de **nodos** continúa en crecimiento.

Para que los robots funcionen con autonomía, es necesario indicarles qué tienen que hacer ante cada situación. Para eso, hace falta compartir un idioma. ¿Cómo nos comunicamos con ellos? La respuesta: mediante lenguajes de programación. Los

robots, en su núcleo, están compuestos de códigos binarios implantados en un disco duro magnético, tal como todas las computadoras modernas. Las instrucciones y datos de un robot están completamente codificados en un sistema de unos y ceros. Cada fragmento de información, sin importar lo complicado que sea, se representa en su totalidad por estos dos números, los cuales también pueden interpretarse como interruptores de “encendido” y “apagado”. Por lo tanto, la mente de un robot está hecha enteramente por billones, incluso trillones, de interruptores, ya sea encendidos o apagados.

En esta secuencia didáctica, se utilizarán servicios en línea alojados en la plataforma IBM Cloud para que TJBot realice más acciones. IBM Cloud de IBM es una plataforma en la nube que permite a desarrolladores, tanto novatos como expertos, crear, desplegar y administrar aplicaciones y servicios. Dentro de la vasta oferta de herramientas alojadas en IBM Cloud, se encuentran los servicios de Watson de IBM.

¿Qué es Watson de IBM?

Watson es una super computadora impulsada por Inteligencia Artificial para resolver problemas por medio de habilidades cognitivas tales como las habilidades cognitivas de un humano: Escucha, habla, observa, siente, razona y toma decisiones.

Las computadoras que hacen uso de Inteligencia Artificial adquieren conocimiento por medio de experiencias, prueba y error, correcciones, tal como un humano lo hace. Lo importante de Watson es que ha sido entrenado por medio de la información contenida en internet en general, y para aplicaciones específicas de Watson en la industria ha sido entrenado por expertos en campos específicos: Medicina, economía, cinematografía, entre otros.

Entrenar a Watson para propósitos específicos (proyectos personales, por ejemplo) también es posible gracias a las herramientas de la nube de IBM, permitiendo así entrenar el reconocimiento visual de Watson para reconocer tipos de alimentos o si es de día o de noche, entrenar a Watson para que establezca conversaciones y atienda las reservaciones de un hotel, entrenar su voz para que pueda hablar de forma entusiasmada o susurrar. Estos son solo algunos de los ejemplos de lo que Watson puede hacer por la industria, por los estudiantes y por las personas, al alcance de todos.

Véase: <https://searchenterpriseai.techtarget.com/definition/IBM-Watson-supercomputer>

<https://searchenterpriseai.techtarget.com/definition/IBM-Watson-supercomputer>

<https://observatorio-ia.com/aplicaciones-de-watson-en-el-sector-sanitario>

¿Qué es el cómputo cognitivo y cómo se relaciona con Watson?

A lo largo de la historia se ha buscado optimizar la manera en la que las computadoras son programadas y adquieren sus conocimientos. Existen diferentes métodos para “enseñar” conocimientos a las computadoras, tradicionalmente el conocimiento se almacena como datos (texto, imágenes, sonido) y un programador por medio de un lenguaje de programación debe dictar la forma en que la computadora debe interpretar estos datos. Dicho de otra forma: El programador debe dictar TODAS las reglas, todos los casos, todas las órdenes a seguir.

Por ejemplo, supongamos que diseñamos una computadora para que haga exclusivamente sándwiches en un restaurante. En el modelo de programación tradicional, tendríamos que programar a la computadora (y generalmente por medio de un lenguaje de programación) los pasos a seguir: Toma una rebanada de pan, untar mayonesa, colocar rebanada de jamón, colocar rebanada de jitomate, colocar rebanada de pan.

Dejamos el sistema funcionando, tres personas aleatorias son escogidas para poner a prueba el sistema: Una de ellas pide un sándwich tradicional, una de ellas sin mayonesa y otra de ellas sin jitomate. El primer sándwich es entregado con éxito, el segundo sándwich nunca es entregado y el tercer sándwich ni siquiera es atendido.

Esto es de esperarse: La computadora fue programada para entregar sándwiches con mayonesa y jitomate, si quieres algo distinto debes ir, acceder al sistema y programar nuevas condiciones.

El ejemplo del sándwich podrá parecer sencillo, pero cuando hablamos de sistemas donde las variantes son exponenciales (Radiografías, control de tráfico) surge la necesidad de una autoadaptación del sistema, un autoaprendizaje que soporte la variabilidad de su entorno y lo haga además en tiempo real: La computación cognitiva ofrece esta versatilidad.

La computación cognitiva imita el funcionamiento humano para aprender cómo lo hacemos nosotros, por medio de experiencias. Para la computadora estas experiencias no son otra cosa más que datos. Entonces los datos se vuelven sumamente importantes, son la fuente de conocimientos de nuestros sistemas, y la forma de obtener estos datos se vuelve sumamente importante, por lo que la computación cognitiva se enfoca nuevamente en cómo los humanos obtenemos nuestros conocimientos: La visión, el oído, el habla, el razonamiento, entre otros medios.

Entonces la computación cognitiva toma forma: Hace uso de métodos cognitivos para obtener información, razona para aprender de estas “experiencias”, y posteriormente hace uso de esos conocimientos para tomar decisiones.

Volviendo al tema del sándwich, ahora colocaremos a nuestra computadora cognitiva a cargo. Se repite la misma tragedia: De los tres primeros sándwiches solo ha podido entregar uno. Pero con este cambio de panorama la corrección del sistema es diferente, en lugar de reprogramar al sistema, la misma computadora busca una

retroalimentación, cuando se le da la instrucción “sin mayonesa” y se le muestra un ejemplo de lo que significa no poner mayonesa, la computadora observará y razonará, aprenderá, y la próxima vez que se le pida no poner un ingrediente la computadora sabrá qué hacer. Esto no requiere intervención técnica, y lo hace de forma automática.

Watson al estar basado en computación cognitiva es la manera en la que ha aprendido de internet en general, de las interacciones entre las personas y de los sensores que rodean el mundo. Los datos los tiene, por lo tanto, los conocimientos del mundo en general los tiene y están a disposición de cualquier persona a través de la nube de IBM.

Véase: <https://www.ibm.com/blogs/transformacion/2016/07/25/los-beneficios-del-computo-cognitivo-para-la-sociedad/>

Servicios de la nube

La nube (también referida como computación en la nube, cloud computing, servicios en la nube) es la manera de referirse a una red mundial de servidores que ofrecen servicios de tecnología de manera remota. Esto es: No es necesario descargar un programa o comprar un dispositivo con la tecnología contenida para poder hacer uso de esos servicios.

Hay nubes para todos tipos de servicio, particularmente IBM cuenta con una nube que ofrece servicios de diferentes índoles: Almacenamiento, Internet de las Cosas, Bases de Datos, e Inteligencia Artificial. Esto último significa que podemos acceder a los servicios de Inteligencia Artificial (Watson) de forma remota, sin componentes complejos ni programas para instalar.

¿Qué es una API?

Una API es una puerta de acceso a una tecnología alojada en internet, esta puerta de acceso brinda información y recibe información. Para comunicarse con una API es necesario tener las credenciales de acceso y la forma adecuada para “iniciar sesión” y pedir la información.

Las ventajas de las APIs es que no requerimos entender cómo funciona el sistema para sus adentros sin importar el tipo de tecnología que use, o la información que provea.

La forma de hacer uso de una API es construir un programa que haga llamadas de este tipo a los sistemas, reciba la información y haga algo con ella. Por ejemplo, podemos construir un pequeño código en un lenguaje de programación que pida el clima actual en nuestra zona. Una API que ofrece esta información es Open Weather Map (<https://openweathermap.org/api>), la API nos daría en el programa la información y por medio de código podemos indicar la instrucción: Si está soleado, apagar luces, en caso contrario, encenderlas.

Hay muchos sistemas que cuentan con APIs para consultar información de ellas, por ejemplo, Twitter tiene una API para que puedas consultar tweets en tiempo real o para

que puedas publicar contenido desde tu propio programa, Spotify cuenta con API para que puedas consultar datos relevantes de las canciones y puedas traerlos directo a tus programas.

Watson cuenta con su propia API, esto significa que podemos construir programas que, con solo credenciales de acceso y peticiones, pueden traer características de Inteligencia Artificial a nuestros programas.

Por último, la ventaja de utilizar llamadas API para obtener información es que pueden ser implementadas en prácticamente cualquier dispositivo con capacidades de procesamiento como una computadora, un celular, un reloj inteligente, entre otros.

Véase: <https://www.nytimes.com/paidpost/ca-technologies/apis-the-building-blocks-of-the-app-economy.html>

¿Qué es un proyecto de código abierto?

Los proyectos de código abierto son proyectos gratuitos que pueden ser modificados por la comunidad, utilizados y distribuidos, de manera que estos pueden crecer exponencialmente y servir a propósitos educativos a escalas masivas.

Véase: <https://opensource.guide/es/start-a-project/>

¿Qué es una Raspberry PI?

Una Raspberry PI es una pequeña computadora del tamaño de una tarjeta de crédito que puede ser programada para diferentes propósitos. Esta tarjeta es complementada por las necesidades del usuario, ya que no cuenta con una pantalla, un mouse o un teclado.

Raspberry PI es muy utilizado en proyectos escolares pues es bastante versátil para experimentar, también es utilizado para integrar sensores y monitorear áreas, puede ser integrado con una cámara y detectores.

La singularidad de Raspberry PI es la variedad de sistemas operativos que soporta, pues puede utilizar Windows, Linux u otros más especializados. En el caso de nuestro taller estaremos utilizando el sistema operativo Raspbian.

Se deja al voluntario la investigación de los componentes de hardware y software que la tarjeta Raspberry PI tiene. En el caso de esta capacitación, estaremos utilizando la tarjeta Raspberry PI 3B+.

Véase: https://www.elconfidencial.com/tecnologia/2013-11-22/dos-millones-de-razones-para-saber-que-es-exactamente-raspberry-pi_56003/

<https://computerhoy.com/noticias/hardware/15-usos-raspberry-pi-que-no-sabias-que-podias-darle-74905>

¿Cómo se trabaja con Raspberry PI?

Como se puede ver, Raspberry PI no cuenta con una pantalla teclado o mouse para interactuar directamente con el sistema. Hay 2 formas de realizar la conexión con el sistema, de forma alámbrica y de forma inalámbrica.

De forma alámbrica es necesario contar con un mouse, teclado, cable HDMI y una pantalla, al encender la Raspberry PI podremos ver el sistema operativo en pantalla e interactuar como si se tratara de un computador común.

De forma inalámbrica tenemos 2 métodos: Por SSH y por VNC. Antes de entrar a detalle es importante conocer dos conceptos: La ventana de comandos y las direcciones IP. Una ventana de comandos es la interfaz de comunicación más básica que cualquier computador tiene, y sirve para interactuar con archivos, configuraciones, instalaciones entre otras cosas sin necesidad de cargar la interfaz gráfica de un sistema. Windows tiene una ventana de comandos que se puede invocar escribiendo “CMD” en el explorador de archivos, Linux cuenta con una ventana de comandos llamada “Terminal”. Ahora, una dirección IP es una serie de números que identifican a un dispositivo conectado al modem de internet, es la manera en que el modem identifica a dónde dirigir los mensajes o las páginas que recibe de internet, para poder enviarlos al dispositivo adecuado. De esta manera, tu computadora o tu celular tienen una dirección IP asignada en el modem.

Volviendo a las conexiones inalámbricas, es importante identificar la dirección IP de la Raspberry PI para que la conexión inalámbrica sepa dónde buscar y poder conectarse. La conexión por SSH es un tipo de conexión que, haciendo uso de la dirección IP, se conecta a la ventana de comandos del dispositivo en cuestión. Esto implica que no tendremos la interfaz gráfica ni navegador de internet, las interacciones serán por comandos directamente en el sistema. Un programa que nos permite realizar esta conexión es “Bitvise SSH” en Windows.

La conexión por VNC, a diferencia de SSH, permite acceder directamente a la interfaz gráfica (como si lo estuviésemos haciendo alámbricamente), y solo necesitamos la dirección IP para realizar la conexión remota. El programa que permite esta conexión se llama “VNC Viewer” para Windows.

Véase: *Raspberry Pi Beginner's Guide: Install and Setup NOOBS:*

<https://www.youtube.com/watch?v=wvxCNQ5AYPg>

Conexión remota VNC: <https://www.youtube.com/watch?v=neoiXYz0jEE>

Conexión remota SSH: <https://www.youtube.com/watch?v=lOP0vwHcdmg>

¿Qué es Node-RED, y en qué lenguaje de programación está basado?

Hemos mencionado que para el uso de una API como lo es Watson generalmente hacemos uso de lenguajes de programación. Hoy en día la programación se ha simplificado de manera que cualquiera pueda hacer uso de estas tecnologías. De ahí nace Node-RED, el cual es una interfaz gráfica que permite programar dispositivos por medio de bloques interconectados. Node-RED es un proyecto de código libre, por lo que miles de usuarios han creado bloques con distintos propósitos, y ofrecen a la comunidad soluciones ya implementadas que pueden servir de referencia para diferentes proyectos.

Los bloques en los que está basado el programa se llaman nodos, los cuales internamente están construidos en un lenguaje de programación llamado JavaScript. Esto permite un nivel de abstracción en dónde los bloques tienen un propósito definido y están encapsulados para no involucrarnos con su programación interna. El conjunto de bloques interconectados entre sí trae soluciones interesantes y poderosas, donde al final la limitante es la imaginación del programador más que la capacidad técnica con la que cuente.

Node-RED cuenta con un buscador de nodos que permite instalar nodos que la comunidad haya creado, entre los que se encuentran nodos de cientos de APIs: Twitter, Facebook, Watson, TJBot, entre otros.

Véase: *Node-RED básico:* <https://www.youtube.com/watch?v=qObGErODKxQ>

¿Qué es el formato de intercambio de información JSON?

Si bien las bases de Node-RED son los nodos, parte importante es la forma en que los nodos se comunican entre sí. Como los nodos pueden hacer uso de tecnologías completamente distintas, la información que se transmitan debe ser clara para el nodo siguiente, y así consecutivamente. Para que exista concordancia entre la información que los nodos se transmiten es necesario un mismo formato, esto es JSON.

JSON es un formato ligero para intercambio de datos que funciona bajo la premisa de llave – valor. Este formato no es otra cosa más que un archivo de texto con la siguiente estructura:

```
Llave de apertura -> {  
    Llave y valor -> "Llave 1" : "Valor 1",  
    Llave y valor -> "Nombre": "Jair",  
    Llave y valor -> "Apellido": "Lizarraga"  
    Llave de cierre -> }
```

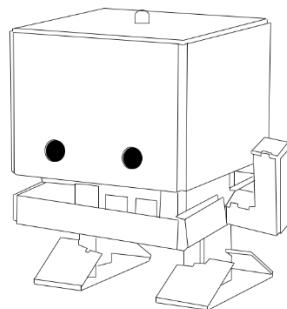
Y para acceder a la información contenida en un JSON, se debe hacer referencia a la llave de la cual deseamos obtener el valor y utilizamos el operador punto para acceder a esa información. Por ejemplo, usando el ejemplo anterior en programación, tenemos el archivo JSON que se llama “Usuario”, la forma para acceder al nombre del usuario sería Usuario.Nombre, o si queremos acceder al apellido del usuario la forma de escribirlo sería Usuario.Apellido.

De esta manera cuando estemos trabajando en Node-RED, si queremos acceder a la información contenida entre los nodos y modificarla o rescatarla, haremos uso del operador punto para acceder a ella.

Véase: https://canela.me/articulo/JSON-JavaScript-jQuery/_¿Qué-es-JSON-para-qué-sirve-y-dónde-se-usa

TJBot

TJBot es un proyecto de código libre que permite al usuario crear su propio robot programable, cuyo propósito es acercar a niños y adultos al mundo de la programación y robótica de una forma fácil y divertida. Por ser de código libre miles de usuarios alrededor del mundo han hecho aportaciones para integrar funcionalidades y nuevos alcances que en un inicio sus creadores jamás habrían vislumbrado.



Haciendo uso de una Raspberry PI, TJBot es capaz de conectarse con los servicios de Inteligencia Artificial de IBM Watson en la nube para adquirir habilidades de computación cognitiva, tales como escuchar, hablar, observar, sentir y aprender, entre otras.

Su programación está basada en Node-RED, por lo que no es requisito conocer un lenguaje de programación para comenzar, y TJBot haciendo uso de sus habilidades cognitivas puede interconectarse con otras APIs disponibles internet para un sinfín de actividades: Conexión con redes sociales, servicios de mensajería, sensores, almacenamiento, entre otros.

La creadora de este robot es Mariam Ashoori, una investigadora e inventora del Centro de Investigaciones de IBM Watson Center, quien a finales del año 2016 en Nueva York

presentó a TJBot en una conferencia y realizó una demostración donde conversaba con el robot. Ese mismo día liberó el código fuente al público en general. Como dato curioso el nombre de TJBot está inspirado en las iniciales del nombre del fundador de IBM, Thomas John Watson,



Mariam Ashoori con TJBot

Preparación del Software

Las actividades desarrolladas para el uso y manejo de TJBot con los participantes requiere que se realice una serie de acciones previas al inicio del taller. Estas instrucciones permitirán inicializar los componentes del robot, y habilitar conexiones remotas a él. En este apartado instalaremos el software necesario para habilitar a un TJBot desde cero y poder realizar una conexión remota con el programa Node-RED de TJBot.

Este procedimiento se realiza solamente una vez por robot, y los componentes necesarios son los siguientes:

- 1 computadora Raspberry Py 3 Model B con cable.
- 1 tarjeta microSD de 16GB de capacidad.
- 1 proyector o monitor con cable HDMI.
- 1 teclado con conexión USB.
- 1 Mouse con conexión USB.

En caso de no disponer de una red WiFi:

- 1 cable ethernet.

Básicamente TJBot reside en la memoria SD, ya que ahí se instalará todo el sistema operativo y las programaciones realizadas, esto significa que podríamos intercambiar prácticamente todos los componentes de TJBot incluida la Raspberry PI, pero conservando la misma memoria SD conservaríamos nuestra programación sin cambio alguno.

Nota:

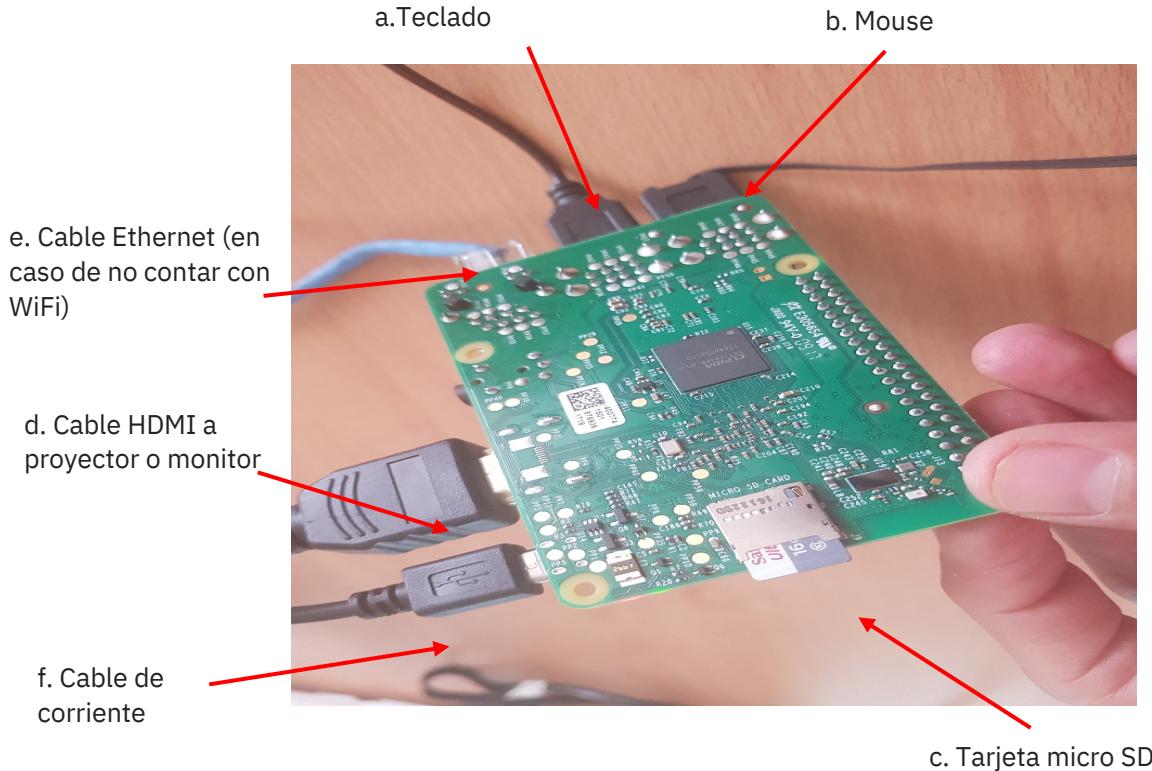
*Se describirá la instalación paso a paso para aquellos que deseen profundizar en los pasos de configuración que se realizan internamente para tener finalmente un TJBot instalado, sin embargo, también puedes realizar una **instalación rápida**, que consiste quemar la memoria SD directamente con una copia de TJBot previamente configurada. Este procedimiento ahorra tiempo y permite la instalación de varios TJBot en poco tiempo. Esto se describe en el apartado **Instalación rápida de TJBot con copia de seguridad**.*

Paso 1: Instalando el sistema operativo Raspbian.

Antes de conectar la computadora Raspberry Pi a la corriente, asegúrese de conectar los siguientes elementos:

- a. Teclado
- b. Mouse

- c. Tarjeta microSD
- d. Cable HDMI a un proyector o monitor
- e. Si no cuentas con red WiFi, conectar un cable LAN.
- f. Cable de corriente (micro USB)



Conecten el eliminador de la Raspberry a la corriente eléctrica.

Nota:

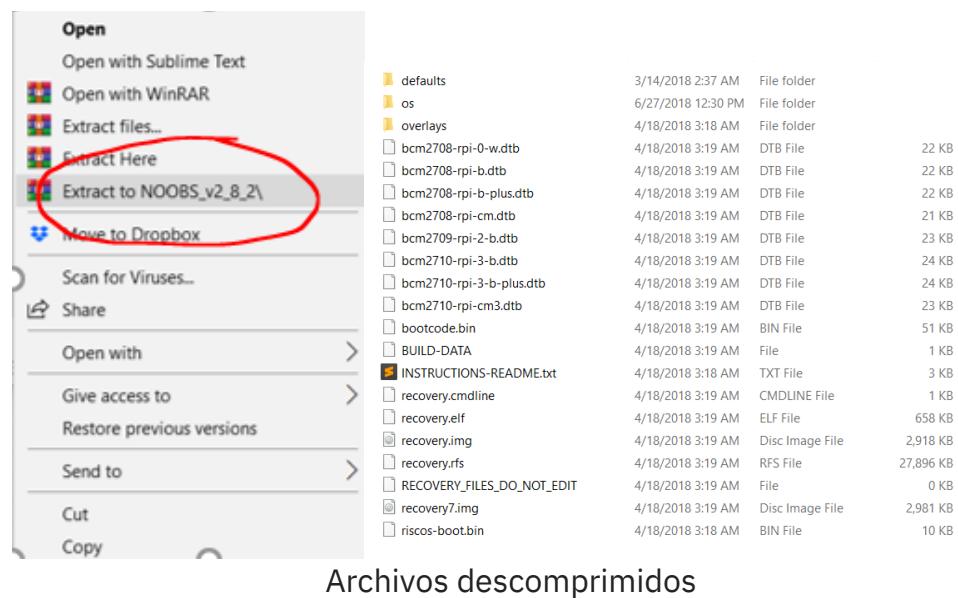
*Si usted **NO** puede ver ninguna imagen en la pantalla, es posible que no estén cargados los archivos de instalación en la memoria, entonces siga las instrucciones “**Si TJBot no muestra imagen en el televisor**”, en caso de que usted **SI** pueda ver imagen en la pantalla, saltarse a “**Si TJBot si muestra imagen en el televisor**”.*

Si TJBot no muestra imagen en el televisor:

Dirigirse a la página: <https://www.raspberrypi.org/downloads/noobs/> y busque el botón de descarga ZIP para NOOBS Desktop:



La descarga comenzará automáticamente. Cuando la descarga haya terminado, diríjase a la carpeta de descargas, y dando click derecho sobre el archivo, de click en “descomprimir en...”.



Nota:

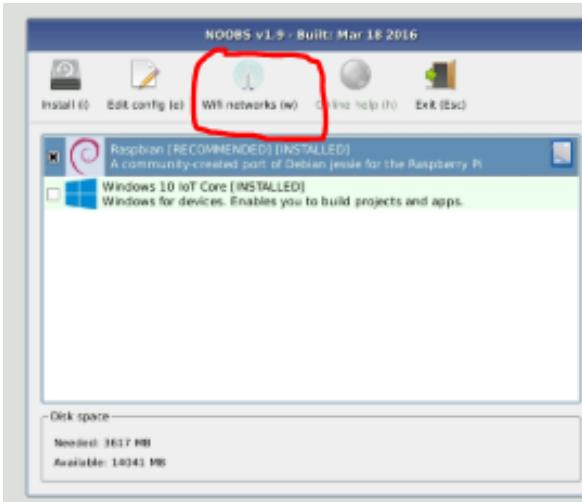
Para descomprimir el archivo que descargó es necesario que su computadora tenga instalado el programa WinRAR (<https://www.winrar.es/descargas>) o similar.

Ahora conecte la memoria microSD a su computadora, entre a la carpeta donde se han descomprimido los archivos y copie los archivos a la memoria SD. Cuando termine de copiar los archivos retire la tarjeta SD de la computadora y conéctela a TJBot.

Conecte a TJBot con el televisor, mouse y teclado, conéctelo a la corriente, y podrá ver en pantalla la ventana de instalación del sistema operativo. Ahora puede continuar con la instalación en el paso 8.

Si TJBot si muestra imagen en el televisor.

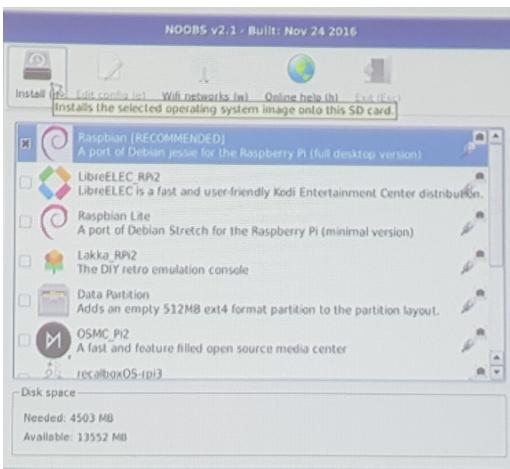
Si usas WiFi agrega tu red dando click en el botón superior izquierdo llamado **WiFi networks**. En caso de que utilices ethernet, no es necesario este paso ni agregar la contraseña.



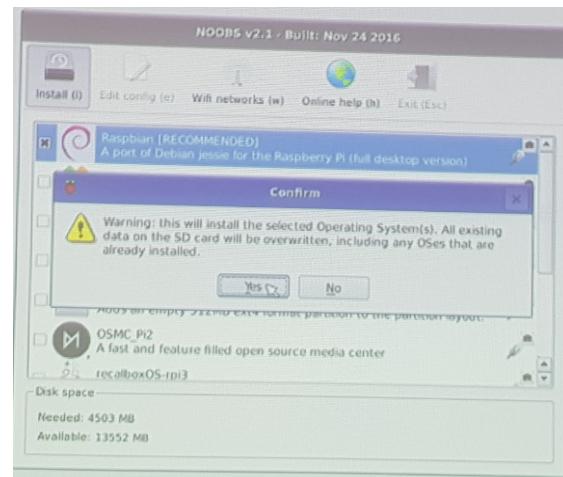
Selecciona tu red, agrega la contraseña en el campo **password** y da click en Ok.



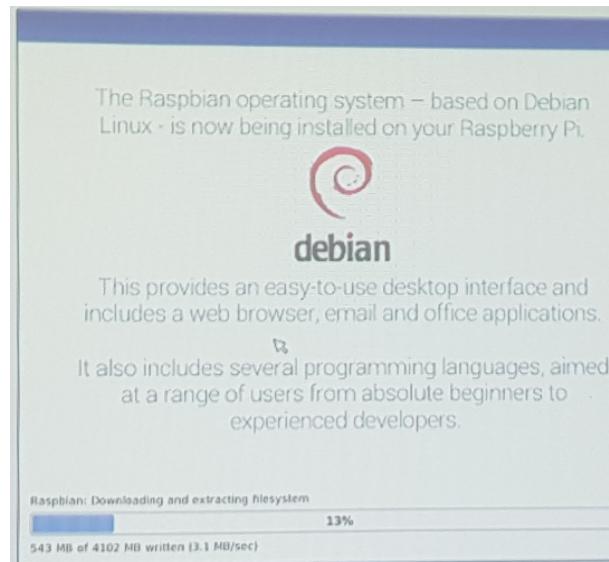
Seleccionen la opción **Raspbian** y de click en el botón de la barra superior llamado **Install**



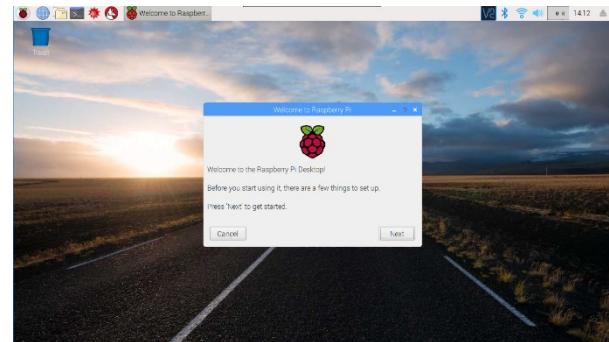
Den click a **Yes** para confirmar la instalación



Se descargarán alrededor de 4 GB.



Al finalizar, la pantalla se verá así:



Da click al botón **Cancel** de la ventana de bienvenida, no nos será de utilidad para nuestra instalación.

Paso 2: Instalando el proyecto TJBot

Dirígete al ícono de frambuesa que se encuentra en la barra superior izquierda de la pantalla, ve a **Preferences**, y luego a **Mouse and Keyboard Settings**. Se abrirá una ventana nueva, abre la pestaña **Keyboard** y selecciona el botón **Keyboard Layout**. En la nueva ventana selecciona como Layout la opción **Spanish** y como variant **Spanish**. Una vez hecho esto, cierra las ventanas dando click en OK.

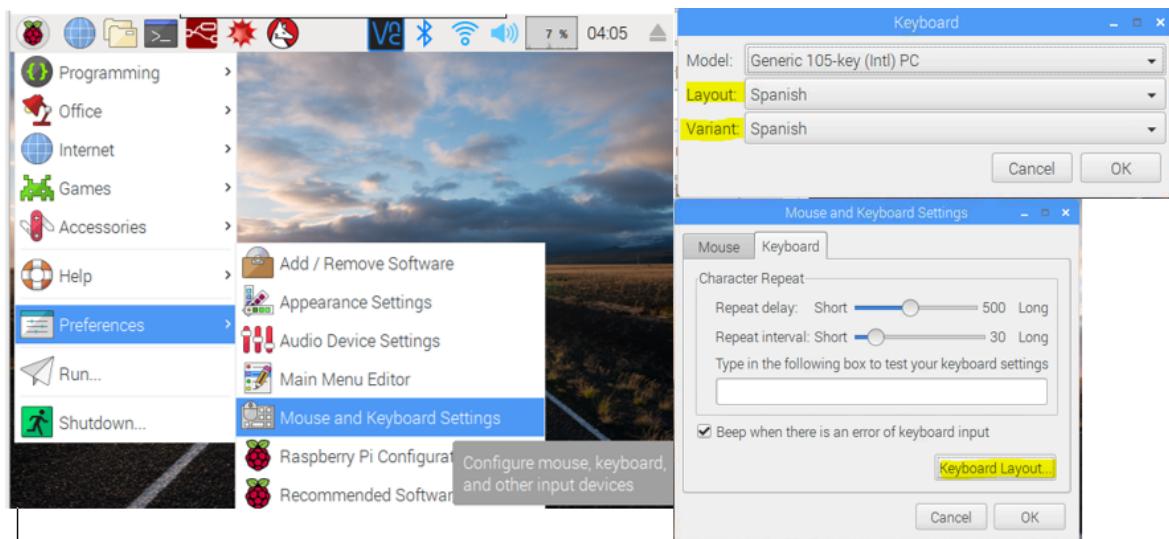


Imagen de ejemplo, podría variar de acuerdo con el sistema operativo.

En la pantalla de inicio abra una ventana de comandos (también llamada terminal) dando click al ícono de la terminal que se encuentra en la barra superior izquierda de la pantalla.

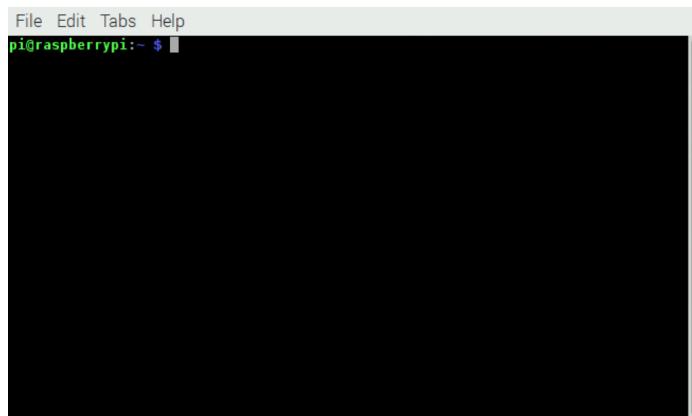


Ícono de terminal



Ubicación del ícono de terminal.

Se abrirá la ventana de comandos:



En esta ventana escribiremos algunos comandos y los introduciremos dando *enter* en el teclado. Preste atención a que los comandos hayan terminado de ejecutarse antes de introducir el siguiente comando, lo sabrá cuando vea el siguiente texto en la ventana:

pi@raspberrypi:~ \$

Escriba el comando **passwd**. Este comando cambiará la contraseña de TJBot. Cuando la ventana de comandos lo indique escribe la contraseña anterior, la cual es **raspberry**, presiona enter y ahora escribe la nueva contraseña **ibmtjbot** y de nuevo enter, vuelve a escribir la contraseña **ibmtjbot** y da enter. Deberá ver el mensaje “Password updated successfully”. De no ser así y recibir algún mensaje de error, repita este paso desde el inicio.

Ahora descargaremos e instalaremos la paquetería del proyecto TJBot con los siguientes comandos:

- Introduzca el comando **git clone <https://github.com/JairLizarraga-ibm/tjbot>**

- b. Introduzca el comando **cd tjbot_setup**
- c. Introduzca el comando **sudo ./bootstrap.sh**

Tras unos minutos el script preguntará si deseas continuar, introduce **y**, luego preguntará si deseas instalar los nodos y nuevamente introduce **y**. La actualización podría tardar hasta 20 minutos en versiones lentas de Raspberry pi. Al finalizar este script la raspberry se reiniciará y la instalación habrá concluido.

Cuando la raspberry reinicie desconecte el mouse, teclado y cable HDMI de la computadora, a partir de este punto las conexiones con TJBot para programarlo serán remotas.

Conexión remota con TJBot: Node-RED

Si la instalación fue exitosa, cada que TJBot encienda comenzará la búsqueda de una red conocida cercana (alámbrica o inalámbrica) para obtener una dirección IP, obtener conexión a internet e iniciar Node-RED.

Presta atención a la información que TJBot va a mencionar, y toma nota de la dirección IP, utiliza la siguiente tabla como referencia:

Prueba exitosa	Prueba fallida
Conectado a la red [Nombre de la red] ó Conexión alámbrica	Buscando red...
Tu dirección IP es [dirección IP]	Buscando red...
Conexión a internet establecida	Error conectando a internet
Servidor iniciado	Error en el servidor Node-RED

Consideraciones que tomar:

- Si TJBot no deja de repetir “Buscando red”, significa que no hay redes conocidas cerca de él.
- Si TJBot encuentra algún error obteniendo la dirección IP u obteniendo conexión a internet, se volverá a ejecutar el análisis hasta que estos errores sean arreglados. Estos errores suelen deberse a la lejanía del modem, o a que la red WiFi no tiene conectividad con internet y no dependen de TJBot para su solución.
- Mientras TJBot no encuentre una red válida con conectividad a internet realizará el análisis cada 5 segundos.
- La dirección IP se encuentra en una serie de 4 números separados por puntos, como este ejemplo: “192.168.100.1”, cada número se debe encontrar en un rango entre 1 a 255.

Nota:

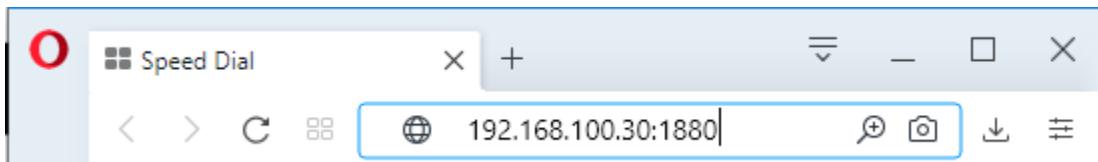
TJBot repetirá la dirección IP dos veces, si no logra escucharla desconecte al robot de la corriente eléctrica y vuelva a conectarlo.

Nota:

Un error que ocurre en ocasiones cuando TJBot anuncia la dirección IP es que dice una dirección IP muy extensa más allá de la serie de 4 números, esto se debe a que TJBot obtuvo una dirección IPv4 y una dirección IPv6 y las dijo en la misma frase. No es necesario anotar todos los números, trate solamente de identificar la primera serie de 4 números teniendo en mente que los números no deben ser mayores a 255.

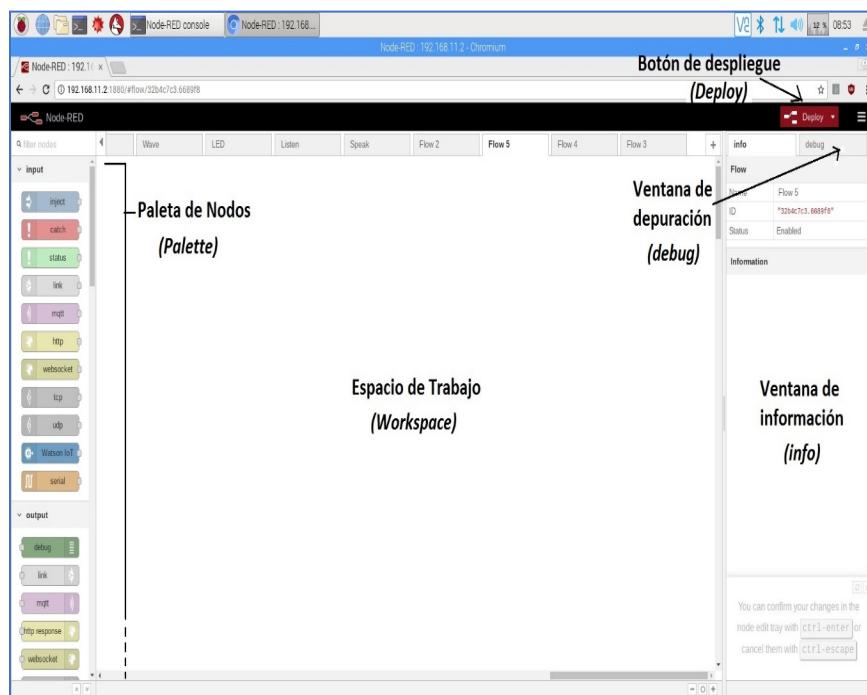
Una vez apuntada la dirección IP del robot escribala en un navegador web desde cualquier dispositivo, ya sea desde su Laptop, o desde TJBot por medio de una pantalla. El dispositivo desde el cual desee conectarse **deberá estar conectado a la misma red que el robot.**

Ya que escribió la dirección en la barra de navegación escriba junto a esta el texto :**1880**, de esta manera indicará al navegador que use el puerto 1880 para entrar al editor de Node-RED.



Ejemplo de la dirección IP de TJBot en el navegador para acceder a Node-RED.

Finalmente introduce la dirección con enter para entrar a la interfaz de programación de Node-RED. Identifiquen los elementos principales del editor de Node-RED a partir de la siguiente imagen:



Interfaz de programación Node-RED.

Verifique nuevamente que la instalación de TJBot fue correcta revisando en la paleta de nodos la existencia del apartado de TJBot, con sus nodos *converse*, *listen*, *see*, *shine*, *speak*, *analyze tone*, *translate* y *wave*.

Agregar redes WiFi a TJBot

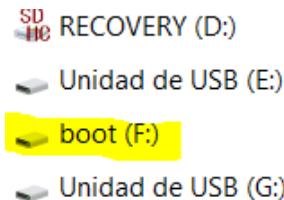
Nota:

Estas **instrucciones** son **válidas** en los sistemas operativos **Windows 10, Linux, Mac OS, Debian, Ubuntu**. Para los sistemas operativos Windows 7 o inferiores, existe la posibilidad de que las particiones no sean leídas correctamente, por lo que se recomienda conectar directamente la raspberry pi a una pantalla para agregar la red WiFi manualmente (Apóyese de la sección Instrucciones para usar el teclado en pantalla).

Al encender, TJBot busca automáticamente una red WiFi ya conocida a la cual conectarse. Cuando desee que TJBot reconozca nuevas redes WiFi puede usted agregarlas por medio de la memoria SD haciendo uso de una computadora o agregarlas a través de una pantalla conectada a TJBot, como lo hizo cuando lo instaló por primera vez.

Para agregar la red por medio de la memoria SD:

1. Conecte la tarjeta microSD de TJBot a una computadora. Si utilizas Windows aparecerán un par de errores de que la memoria no puede leerse correctamente, ignoralos.
2. En su computadora diríjase a “Mi PC” para ver los nuevos dispositivos conectados. Verá que se agregaron 4 nuevas particiones. Seleccione la partición llamada boot. Si esta partición no aparece significa que estás usando un sistema operativo incompatible con este proceso, revisa la nota inicial.



Particiones detectadas en la computadora.

3. Dentro de esta partición encontrará una serie de archivos, con un editor de textos abra el archivo llamado *mi_red_WiFi.txt*. Deberá ver el siguiente contenido.

```

2   network={
3       ssid=""
4       psk=""
5       key_mgmt=WPA-PSK
6   }

```

Es importante que modifique este archivo con cuidado, ya que si escribe caracteres fuera de los espacios indicados podría corromper la configuración WiFi y esta quedar deshabilitada. Siga atentamente los pasos siguientes según sea su caso:

Si la red...	
Cuenta con contraseña	No tiene contraseña (abierta)

Escribe entre las comillas del campo ssid el nombre de la red, incluidas mayúsculas, minúsculas, números y signos de puntuación, y en el campo psk, deberá escribir la contraseña, asegurándose de que la contraseña es la correcta.

Escribe el nombre de la red a la que desea conectarse, eliminar el renglón “psk” y en key_mgmt escribir **NONE**. Use el siguiente ejemplo como referencia.

Ejemplo

```
network={  
    ssid="MiRed"  
    psk="MiContraseña"  
    key_mgmt=WPA-PSK  
}
```

Ejemplo

```
network={  
    ssid="MiRed"  
    key_mgmt=NONE  
}
```

4. Ahora retire la tarjeta SD, colóquela en TJBot y conéctelo a la corriente. Escuchará que el TJBot detecta la nueva configuración y se reiniciará automáticamente. Si la configuración fue exitosa, 40 segundos después de reiniciarse escuchará que TJBot se ha conectado a la red configurada.

Nota:

Si tras realizar los pasos, se pierde conectividad a internet y TJBot no vuelve a conseguir conectarse a ninguna red WiFi, es posible que haya modificado erróneamente los parámetros anteriores en el archivo de texto, esto conllevará. Para corregir esto:

- Conecte nuevamente la tarjeta SD a la computadora, diríjase a la partición boot y abra de nuevo el archivo **mi_red_WiFi.txt**.
- Borre todo el contenido del archivo dejándolo en blanco, y en la primera línea escriba la palabra **restart**.
- Expulse la tarjeta SD e introduzcala a TJBot, enciéndalo.
- TJBot detectará el cambio y dejará en blanco su listado de redes conocidas.
- TJBot habrá habilitado nuevamente el Wifi, ahora agregue nuevamente una red a TJBot.

Instrucciones para usar el teclado en pantalla

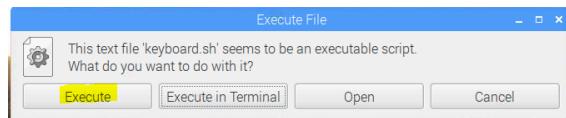
En ocasiones es más cómodo para el usuario agregar redes WiFi a través de la interfaz gráfica, para esto podemos hacer uso del teclado en pantalla que viene integrado en la Raspberry. Para el proceso necesitaremos conectar la Raspberry a una pantalla y un mouse.

Conecta a la raspberry el mouse y la pantalla, ahora enciende a TJBot. Una vez que inicia. Cuando la interfaz gráfica haya iniciado dirígete al ícono **Keyboard.sh** y da doble click en él para ejecutarlo.

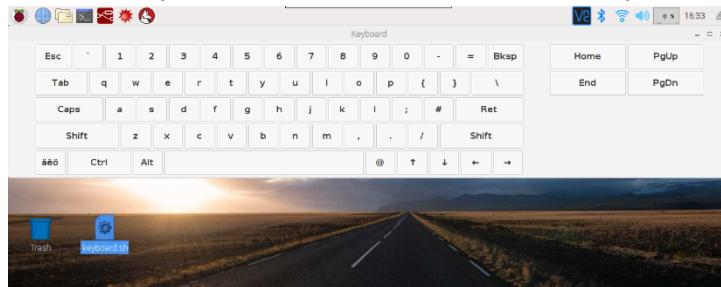


Ícono del teclado en pantalla.

5. Ahora selecciona **Execute** para abrir el teclado.



Finalmente podrás hacer uso del teclado en pantalla.



Con el teclado en pantalla podrás escribir prescindiendo del teclado físico donde sea que te encuentres. Prueba dando click en el ícono de WiFi en la esquina superior derecha y seleccionando la red a la que deseas conectarte.

Instalación rápida de TJBot con copia de seguridad

Podemos instalar rápidamente copias preconfiguradas de TJBot en la memoria SD para evitar la instalación paso a paso, esto resulta muy útil a aquellas personas que no tengan interés en involucrarse con los componentes internos de software o que deseen configurar muchos TJBots de forma rápida.

En la página <https://cognitiveparatodos.mybluemix.net/tutoriales/tjbot> podrás encontrar una copia (imagen) preinstalada de TJBot que el equipo de IBM ha puesto a tu disposición para que puedas utilizar en tu robot.

El proceso para instalar una imagen de TJBot es la siguiente:

1. Descarga el archivo que contiene la imagen a grabar.
2. El archivo se descargará en formato ZIP, por lo que debes descomprimirlo con un programa de compresión/descompresión de archivos como WinRAR (<https://www.winrar.es/descargas>).

En caso de que desees saltarte toda la instalación manual, puedes simplemente grabar este archivo en tu memoria SD:

- Link de Google Drive: [Descargar](#)

El archivo pesa aproximadamente 3Gb por lo que la descarga podrá tardar algunos minutos. Una vez descargado el archivo te recomendamos alguno de los siguientes programas para grabarlo en tu memoria SD:

Botón de descarga.

3. Instala alguno de los siguientes programas de grabado de memorias SD:

- a. Para Windows: Win32 Disk Imager:
<https://sourceforge.net/projects/win32diskimager/>
 - b. Mac y Linux: Etcher
<https://etcher.io>
4. Aunque las instrucciones varían de acuerdo al programa instalado, el proceso es similar: Conecta la memoria SD a la computadora, selecciona en el programa la memoria SD conectada, selecciona el archivo imagen que descargaste y da click en “Write”.
 5. El proceso demora alrededor de 15 minutos, cuando concluya extrae la memoria SD de la computadora.
 6. Conecta la memoria SD a TJBot y comprueba que la instalación fue exitosa.

Nota:

Puedes grabar más de una memoria SD de forma simultánea, solo abre de forma simultanea el programa de grabado de memorias.

Formatear o reinstalar un TJBot ya grabado

Durante la práctica han ocurrido casos en que la instalación de TJBot fue errónea, o que TJBot presenta comportamientos defectuosos como con la conectividad de internet, malas configuraciones o programaciones que eliminaron o movieron archivos importantes para el funcionamiento normal, cualquiera que sea el caso el usuario deseará simplemente comenzar desde cero con un TJBot nuevo. En el caso de usuarios con computadoras Linux o Mac formatear la memoria no es difícil y basta con un programa de formateo de memorias tradicional para llevar a cabo esta tarea. En caso de usuarios con sistemas operativos Windows en sus computadoras esto es más problemático. Para entender esta complicación debemos entender la estructura interna de la memoria SD de TJBot.

Como se ha mencionado anteriormente TJBot utiliza un sistema operativo llamado Raspbian, un derivado de Linux. Linux es un sistema operativo cuyo funcionamiento interno es distinto a Windows, y la diferencia principal es la estructura de archivos interna que crea para funcionar, con formatos incompatibles para Windows.

Cuando nosotros grabamos por primera vez una memoria en blanco con el proyecto TJBot estamos automáticamente creando una estructura de datos preparada para Linux, por tanto, incompatible parcialmente con Windows. Se crean cuatro particiones (secciones) en la memoria, algunas de estas particiones se encuentran en un formato

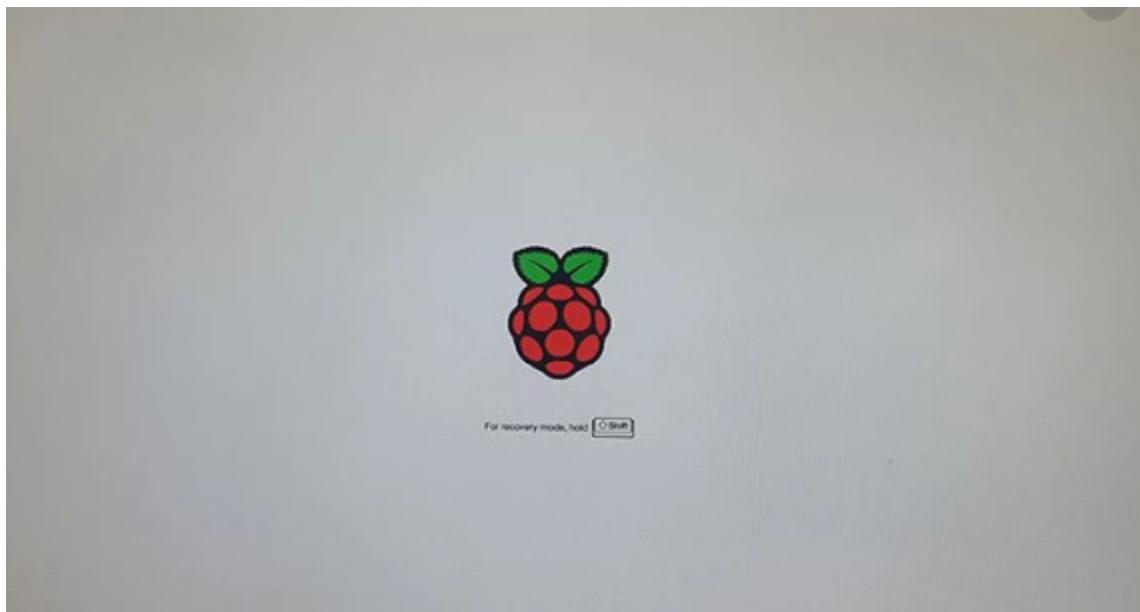
incompatible para Windows como lo son EXT4, y algunas de ellas en un formato compatible como NTFS Y FAT.

Entendiendo esto la problemática radica en que formatear la memoria con Windows es imposible debido a la incompatibilidad de sistemas operativos: Una vez quemada la memoria, Windows no podrá leer ni manipular correctamente las particiones en formato EXT. Entonces... ¿Cuál es la solución?

Para usuarios Linux o Mac ya está dicho, un simple programa de formateo bastará para dejar la memoria en blanco, de hecho bastará un formateo rápido para que la memoria esté lista para ser grabada nuevamente.

Para usuarios Windows no hay alternativa mas que la resignación de acudir a una computadora con otro sistema operativo, o instalar a TJBot manualmente desde el sistema operativo como se puede leer en las *instrucciones para instalar el sistema operativo Raspbian* pero con una pequeña diferencia:

1. Conectar a la Raspberry PI un mouse, teclado y pantalla o proyector.
2. Conforme el sistema operativo inicia se mostrará en pantalla solo por unos segundos una imagen de una frambuesa, con la leyenda “For recovery mode, press shift”, mantén presionado shift en el teclado.



Pantalla para entrar al modo de recuperación.

3. Tras esto entrarás al menú de recuperación de la Raspberry, que nos dará la opción de instalar desde cero un sistema operativo. Llegado a este punto deberás

seguir las instrucciones para instalar desde cero a TJBot ya que si continuas, perderás todos los datos en tu memoria SD

Construyendo a TJBot

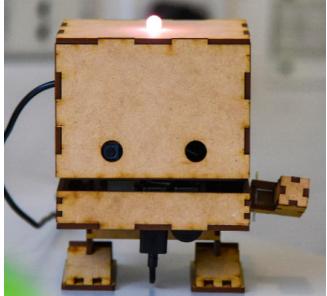
Los robots requieren una serie de elementos acordes a la actividad particular para la que fueron creados. En esta lección observaremos todos los componentes por separado y después realizaremos el armado.

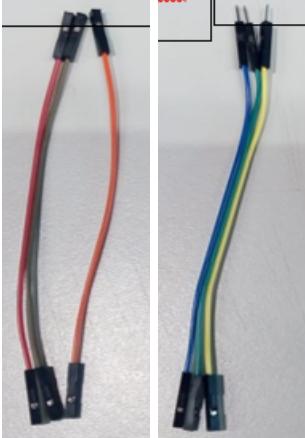
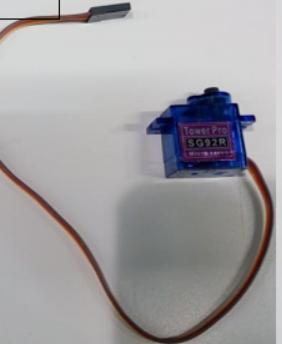
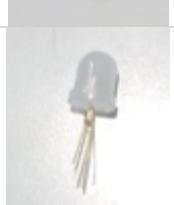
Objetivos

- Conocer los componentes de TJBot.
- Construcción de la carcasa.
- Integración de la carcasa con los componentes electrónicos.
- Conexión de los componentes electrónicos.

Material necesario por TJBot

Los siguientes componentes integran en su totalidad a TJBot. Actualmente la mayoría son sencillos de conseguir en tiendas de electrónica convencionales, si bien algunas tiendas lo venden como un kit:

Nombre	#	Ejemplo
Estructura de TJBot desarmada (MDF, impresión 3D o cartón).	1	
Tarjeta Raspberry Py 3 Model B o superior con cable micro USB o cable de alimentación micro USB.	1	

Mini micrófono USB.	1		
Cables jumper hembra/hembra de colores preferentemente negro, amarillo y verde. Cables jumper hembra/macho de colores preferentemente café, rojo y naranja.	3 y 3		
Servo motor.	1		
Tarjeta microSD de 16GB de capacidad.	1		
Foco LED (neoPixel Diffused 8mm LED-x5).	1		
Cámara para Raspberry Pi NoIR Camera V2 o superior.	1		

Bocina USB.	1		
Fuente de alimentación	1		

Construcción de la carcasa de TJBot

Actualmente existen 3 tipos de materiales con los cuales la carcasa de TJBot puede ser construida, la selección del material deberá ser tomada de acuerdo con el uso que se tenga en mente, el precio y la disponibilidad de realizar el corte del material en la región donde te encuentres.

Los planos para cada uno de los materiales e instrucciones más detalladas pueden ser obtenidas de la siguiente liga:

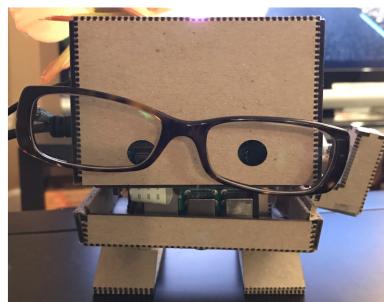
<https://cognitiveparatodos.mybluemix.net/tutoriales/tjbot>

Para poder hacer uso de estos planos debes ir a un lugar donde hagan cortes de materiales (para el caso de cartón y MDF) o donde hagan impresiones 3D, y utilizar los archivos descargados para que una computadora los realice por ti. **Estos archivos no son una guía** para hacer una carcasa a mano, debido a que los planos requieren precisión de milímetros para poder encajar correctamente.

- Cartón

El cartón fue el primer material con el cual fue creado el robot. Consiste en un cartón cortado por laser capas de ser doblado para tomar la forma que se necesita. La rigidez del robot dependerá de la fortaleza del cartón seleccionado, por lo que se recomienda mucho cuidado con el armado en cartón.

Las instrucciones del armado puedes encontrarlas en el video **TJBot Assembly**:
<https://youtu.be/bLt3Cf2Ui3o>



TJBot en cartón

— Impresión 3D

La impresión 3D propone un material impreso usando plástico para crear cada una de las piezas de TJBot. Estas piezas ya están “dobladitas”, así que las instrucciones del armado son las mismas que las del video **TJBot Assembly**, pero a partir del segundo 39: <https://youtu.be/bLt3Cf2Ui3o?t=39>



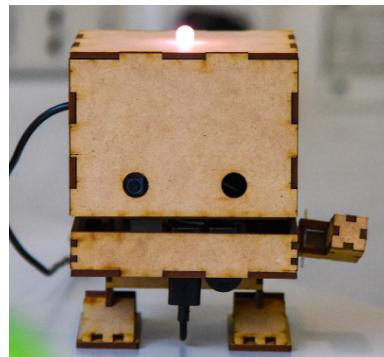
TJBot en impresión 3D.

— MDF

El MDF son fibras de madera comprimida mucho más resistente que el cartón, el cual resiste golpes y caídas, se ha convertido en nuestro material más utilizado por su bajo costo y su alta resistencia, además que puede ser coloreado por los alumnos. Su armado es distinto al cartón o la impresión debido a que este material no puede ser doblado.

Utiliza este video para llevar a cabo el armado con este material:

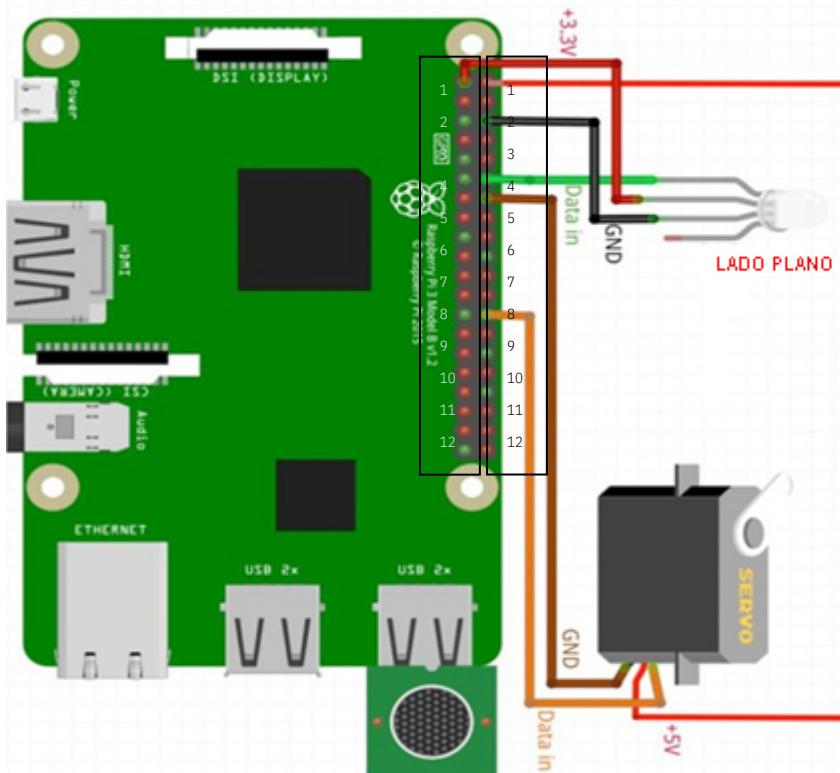
<https://youtu.be/Yee2ylksP4U>



TJBot en MDF

Los videos no son muy descriptivos al hablar de las conexiones de los componentes electrónicos, por lo que deberás hacer uso de las siguientes instrucciones:

1. Conexión del servomotor y el led con la Raspberry.



LED

Rojo	Izquierda - 1
Negro	Derecha - 3
Verde	Derecha - 6

Motor

Cafe	Derecha - 7
Amarillo	Derecha - 13
Rojo	Derecha - 1

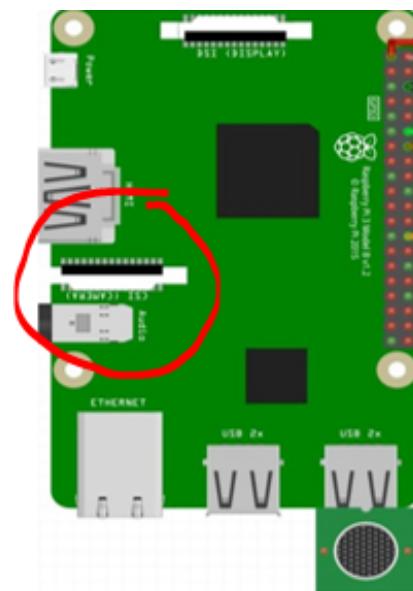
Conectando el LED.	Conectando el motor.
Utilizando los cables hembra hembra, conectaremos el Led hacia la tarjeta Raspberry Pi. Hay que notar que el Led tiene un lado plano en su plástico, el cual puedes identificar tocándolo con tus dedos. Los cables que utilizaremos para conectar el Led serán los que se encuentren en la parte posterior de la tarjeta.	Para el motor, tomaremos los cables hembra macho, los cuales unirán los los pines de la tarjeta Raspberry Pi a los cables del motor. En caso de que los colores del motor no coincidan con el del

dedos o mirándolo a contraluz. Este lado plano nos servirá de referencia para la conexión de los cables, por lo tanto, la patita que está del lado plano del Led no lleva conexión, la patita al lado de esta usa un cable negro, luego uno rojo y al final el verde.

diagrama, prestaremos atención que la conexión del cable rojo del diagrama corresponda con el cable rojo del motor, que la conexión del cable café del diagrama vaya con el cable negro del motor, y por último el cable blanco del motor vaya con el cable naranja del diagrama.

2. Conexión de la cámara.

Para conectar la cámara, buscaremos en la Raspberry una conexión llamada “CAMERA”, la cual consiste en una pequeña bandeja que deberás jalar hacia fuera ligeramente, tomarás la cámara por la cinta, y te asegurarás de que la parte azul apunte hacia abajo (usa la imagen de ejemplo como referencia), introduce hasta topar la cinta, y luego devuelve la bandeja a su posición original.



Conexión de la cámara, con los puertos USB apuntando hacia abajo

3. Por último, deberá conectar el micrófono y la bocina a cualquier puerto USB.

¡TJBot está listo!



Programando con TJBot

La programación con TJBot es sencilla gracias a la interfaz gráfica Node-RED, gracias a la cual no necesitamos un lenguaje de programación para construir actividades y algoritmos interesantes y poderosos. Node-RED puede instalarse en una variedad de dispositivos y sistemas operativos para que estos sean programados, como lo es el caso de la Raspberry PI de TJBot.

Conexión remota a TJBot con Node-RED

A diferencia de un programa tradicional de computadora, Node-RED es más bien un servidor que se inicia en el dispositivo, y para poder usarlo es necesario visitar el navegador web y entrar a la página que el servidor ha habilitado. Cuando el servidor es iniciado, se habilita una página web disponible a nivel local que, en lugar de ser una dirección web típica como www.ibm.com, resulta ser la dirección IP del robot junto con un puerto de funcionamiento.

La dirección IP es la localización del TJBot dentro de la red de internet a la que está conectado, de esta forma los dispositivos de esta red pueden conectarse al servidor Node-RED. Las direcciones IP a través de las cuales podemos conectarnos al servidor Node-RED de TJBot tienen un formato de cuatro series de números que cada uno está dentro de un rango desde 0 hasta 255, por ejemplo “192.168.100.1”.

Entonces es importante destacar que:

1. Cualquier dispositivo que esté conectado a la misma red que TJBot puede entrar a Node-RED y programarlo, por ejemplo celulares, tabletas y computadoras.
2. Si la computadora se encuentra en una red o un segmento de red diferente de TJBot, no podrá tener acceso a TJBot.

Podemos concluir entonces que la conexión con TJBot es remota, no hay un cable entre la computadora y el robot.

Para realizar una conexión remota con TJBot enciéndalo cerca de una red conocida para el robot, a los 30 segundos TJBot comenzará la búsqueda de una red conocida cercana (alámbrica o inalámbrica) para obtener una dirección IP, obtener conexión a internet e iniciar Node-RED.

Presta atención a la información que TJBot va a mencionar, y toma nota de la dirección IP, utiliza la siguiente tabla como referencia:

Prueba exitosa	Prueba fallida
Conectado a la red [Nombre de la red] ó Conexión alámbrica	Buscando red...
Tu dirección IP es [dirección IP]	Buscando red...
Conexión a internet establecida	Error conectando a internet
Servidor iniciado	Error en el servidor Node-RED

Consideraciones que tomar:

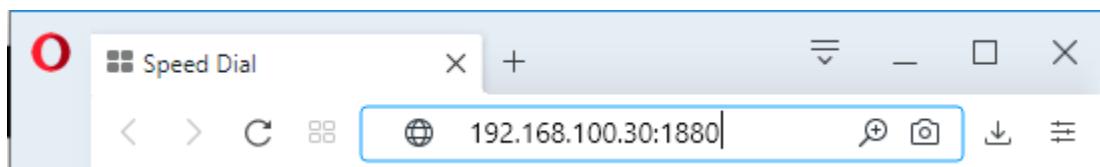
- Si TJBot no deja de repetir “Buscando red”, significa que no hay redes conocidas cerca de él.
- Si TJBot encuentra algún error obteniendo la dirección IP u obteniendo conexión a internet, se volverá a ejecutar el análisis hasta que estos errores sean arreglados. Estos errores suelen deberse a la lejanía del modem, o a que la red WiFi no tiene conectividad con internet y no dependen de TJBot para su solución.
- Mientras TJBot no encuentre una red válida con conectividad a internet realizará el análisis cada 5 segundos.
- La dirección IP se encuentra en una serie de 4 números separados por puntos, como este ejemplo: “192.168.100.1”, cada número se debe encontrar en un rango entre 1 a 255.

Nota:

1. *TJBot repetirá la dirección IP dos veces, si no logra escucharla desconecte al robot de la corriente eléctrica y vuelva a conectarlo.*

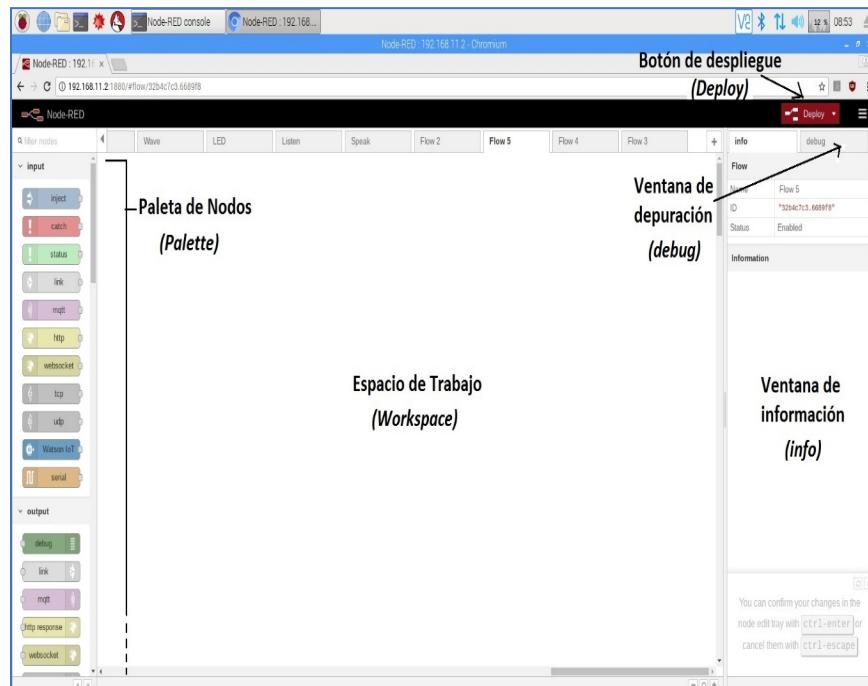
2. Un error que ocurre en ocasiones cuando TJBot anuncia la dirección IP es que dice una dirección IP muy extensa más allá de la serie de 4 números, esto se debe a que TJBot obtuvo una dirección IPv4 y una dirección IPv6 y las dijo en la misma frase. No es necesario anotar todos los números, trate solamente de identificar la primera serie de 4 números teniendo en mente que los números no deben ser mayores a 255.

Una vez apuntada la dirección IP del robot escríbala en un navegador web y junto a esta el texto :**1880**, de esta manera indicará al navegador que use el puerto 1880 para entrar al editor de Node-RED.



Ejemplo de la dirección IP de TJBot en el navegador para acceder a Node-RED.

Finalmente introduce la dirección con enter para entrar a la interfaz de programación de Node-RED.



Interfaz de programación Node-RED.

Instrucciones para crear cuenta en IBM Cloud

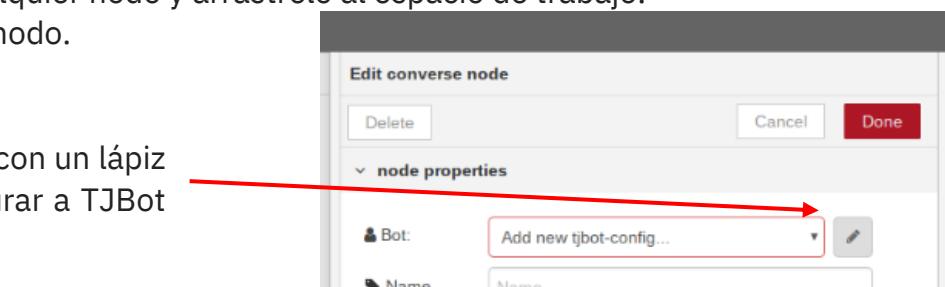
Antes de comenzar la programación vamos a asegurarnos de tener las credenciales de IBM Watson necesarias para configurar los servicios de inteligencia artificial en TJBot, para esto es necesario crear una cuenta en la nube de IBM, crear los servicios en la nube y obtener las credenciales de API.

1. En un navegador WEB abra la página de IBM Cloud: <https://cloud.ibm.com/login>
2. Dé click en el botón “Create an IBM Cloud account”.

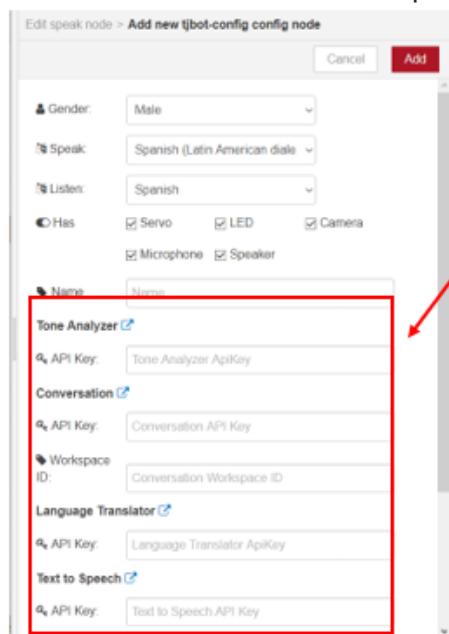
3. Llene los datos que le solicitan y espere un correo electrónico para confirmar su registro.
4. En el correo electrónico que reciba dé click en el botón “Confirm Account”. Revise tu bandeja de spam en caso de no recibir el correo.
5. Inicie sesión en IBM Cloud con el correo electrónico que registró y su contraseña, luego de esto verá la pantalla Dashboard.

Configurando a TBot en Node-RED

1. Estando en la interfaz Node-red identifique en la barra lateral izquierda la sección “TJBot”. Elija cualquier nodo y arrástrelo al espacio de trabajo.
2. Dé doble click al nodo.
3. Dé click al botón con un lápiz para para configurar a TJBot
4. Seleccionen el género de TJBot (puede ser masculino o femenino), los idiomas que habla y escucha (Spanish North American) y seleccionen la casilla de todos los elementos que tiene (Servo, LED, Camera, Microphone, Speaker). También escriban TJBot en el espacio para el nombre.
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.



6. Den click al cuadrado con una flecha al lado de la sección “Text to speech” para abrir la página de IBM Cloud.



The screenshot shows the 'Text to Speech' service configuration page. On the left, there's a sidebar with 'Text to Speech' highlighted. The main area displays service details like 'Service Name: Text to Speech-1' and lists supported languages and features. At the bottom right, there's a large blue 'Create' button.

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

This screenshot shows the pricing page for the Text-to-Speech service. It includes plans for 'Standard' and 'Advanced' with their respective costs. At the bottom right, there's a blue 'Create' button.

8. En la página que se abrió, en el menú izquierdo asegurate de encontrarte en la sección de gestión o “Manage”.

This screenshot shows the 'Manage' section of the IBM Cloud interface. The sidebar has 'Gestionar' (Manage) highlighted with a red circle. Other options like 'Iniciación' (Get Started), 'Credenciales de servicio' (Service Credentials), 'Plan', and 'Conexiones' are also visible.

9. En esta página da click en el botón “show” para mostrar las credenciales. Selecciona el “API Key” y peguelo en la sección correspondiente del editor de Node-Red.

This screenshot shows the Node-RED editor with the 'Credentials' tab selected. In the center, a configuration dialog for a 'Text to Speech' node is open. The 'Add' button is highlighted with a red box. Below it, the 'Text to Speech API Key' field contains the copied API key, which is also highlighted with a red box.

10. Repite los pasos del 6 al 9 para los servicios de Habla a texto (Speech to Text), Texto a Habla (Texto to speech), Reconocimiento visual (Visual recognition), Traductor de lenguaje (Language Translator), Tone Analyzer y Conversation.
11. Una vez añadidas las credenciales, da click en “Add” o “Update” y luego en Done.

Mi primer programa

En nuestro primer programa vamos uno de los conceptos más importantes para entender la programación con Node-RED, el formato de intercambio de información JSON.

Como se mencionó en los antecedentes, JSON es un formato sencillo por medio del cual podemos transmitir información entre computadoras, consta básicamente de una llave de apertura y cierre que contiene dentro llaves y valores separados por comas.

```
{  
  "Id": 0,  
  "FirstName": "string",  
  "LastName": "string",  
  "Name": "string",  
  "EmailAddress": "string",  
  "TerritoryId": 0  
}
```

Ejemplo de un JSON.

Para acceder a la información contenida en un JSON se utiliza el operador punto. Podemos tomar como ejemplo la imagen anterior, suponiendo que la información se encuentra contenida en una variable llamada “msg” (haciendo referencia a *message* o mensaje del inglés), para acceder al dato “Id” de nuestro JSON la operación a realizar sobre este elemento es: msg.Id.

Es importante familiarizarse con el manejo de los JSON debido a que, la comunicación entre los nodos de Node-RED se hace completamente en un formato JSON, aquí cada nodo toma la información que necesita, la procesa, y la vuelve a alojar en ese objeto JSON (dentro de la aplicación, el objeto tiene por nombre msg).

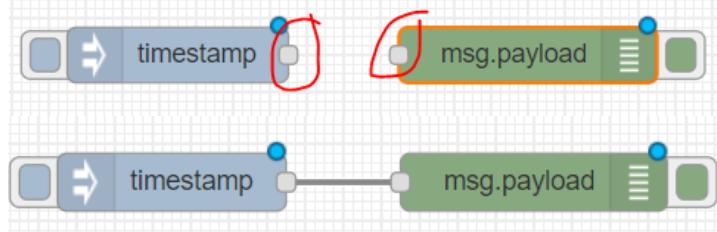
Gracias a que los nodos se encuentran preprogramados, la mayor parte del trabajo que tendremos que realizar será la de configurar cada uno de los nodos para que funcione como deseamos, y orquestar la información contenida en el objeto msg que se está transmitiendo, esto es, alojar u obtener la información importante de la llave correcta.

El primer programa que realizaremos nos mostrará el JSON que los nodos transmiten, las llaves y valores que inicialmente contienen y manipularemos la información de una de sus llaves.

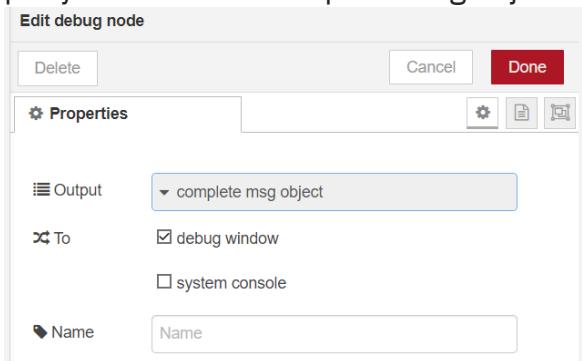
1. Conéctate a la interfaz de Node-RED de tu TJBot.
2. Del lado izquierdo se encuentra un buscador de nodos, que evitará que busques manualmente cada nodo. Busca los nodos “Inject” y “Debug”, arrastra cada uno de ellos al espacio de trabajo



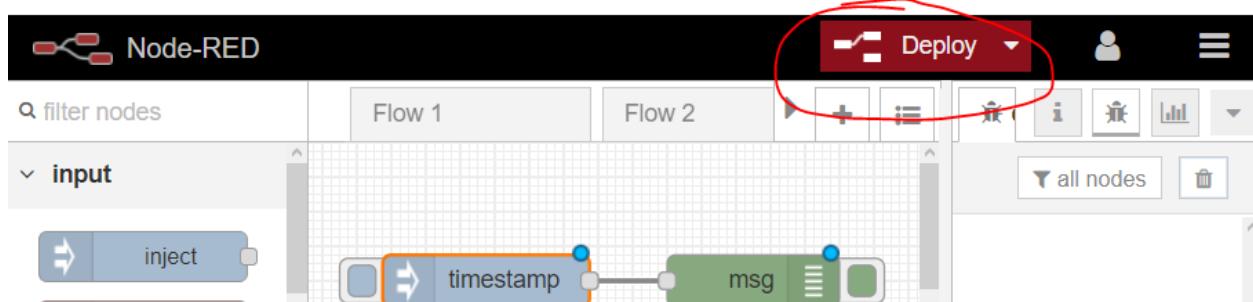
3. Une los nodos por medio del punto gris.



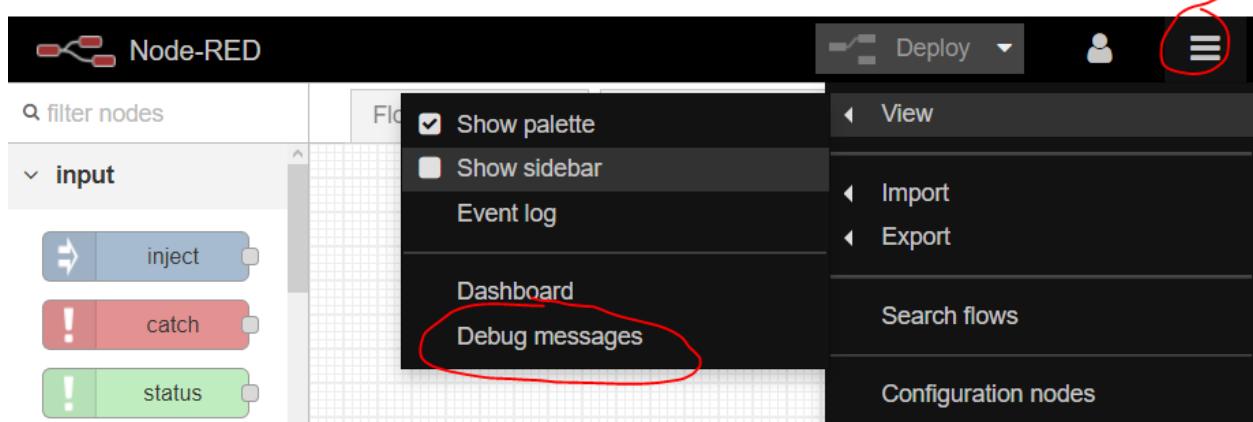
4. Da doble click al nodo debug para configurarlo. En esta ventana despliega las opciones de Output y selecciona “Complete msg object”. Da click en done.



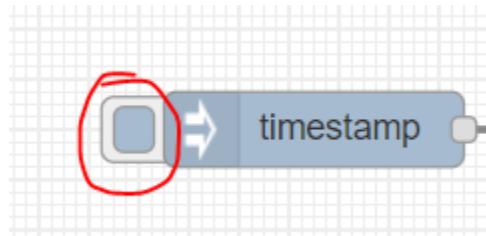
5. Da click a un botón que se ha iluminado en la esquina superior derecha llamado "Deploy" que sirve para guardar los cambios.



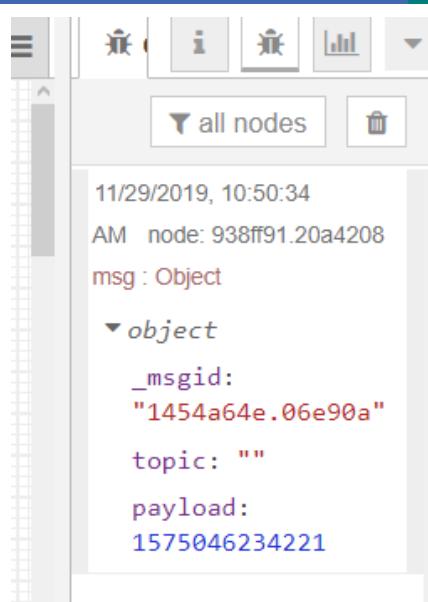
6. Un mensaje que dice "Successfully deployed" anunciará que el guardado fue exitoso.
 7. En el menú superior derecho busca la opción View y luego la opción Debug messages. Del lado derecho se abrirá una ventana en blanco.



8. Inicia la actividad dando click al botón azul que se habilitó al costado del nodo inject.



9. Se desplegará un mensaje del lado derecho, da click una vez en el para poder verlo de forma ordenada. Este mensaje es el JSON que el nodo Inject emite.



10. Ahora da doble click al nodo Inject para modificarlo. La opción que configuraremos será Payload, da click en el menú desplegable y selecciona “string”, en el campo que se ha abierto escriba un mensaje.

Edit inject node

Properties

Payload: timestamp

Topic:

Inject once after 0.1 seconds, then

Repeat: none

Name: Name

Note: "interval between times" and "at a specific time" will use cron.
"interval" should be less than 596 hours.
See info box for details.

Edit inject node

Properties

Payload: **Hola** (circled in red)

Topic:

Inject once after 0.1 seconds, then

Repeat: none

Name: Name

Note: "interval between times" and "at a specific time" will use cron.
"interval" should be less than 596 hours.
See info box for details.

11. Nuevamente guarde los cambios con deploy e inicie la actividad para ver el siguiente resultado en la ventana debug.

```

AM node: 938ff91.20a4208
msg : Object
  ▼ object
    _msgid:
    "877aacd2.73a11"
    topic: ""
    payload: "Hola"

```

Esta es la primera actividad básica para comprender el funcionamiento interno de los nodos y flujos que programaremos a lo largo del manual.

Los dos primeros nodos que hemos utilizado son Inject y Debug.

Inject sirve para iniciar una actividad, transmite el primer JSON llamado “msg” al resto de los nodos que están conectados a él.

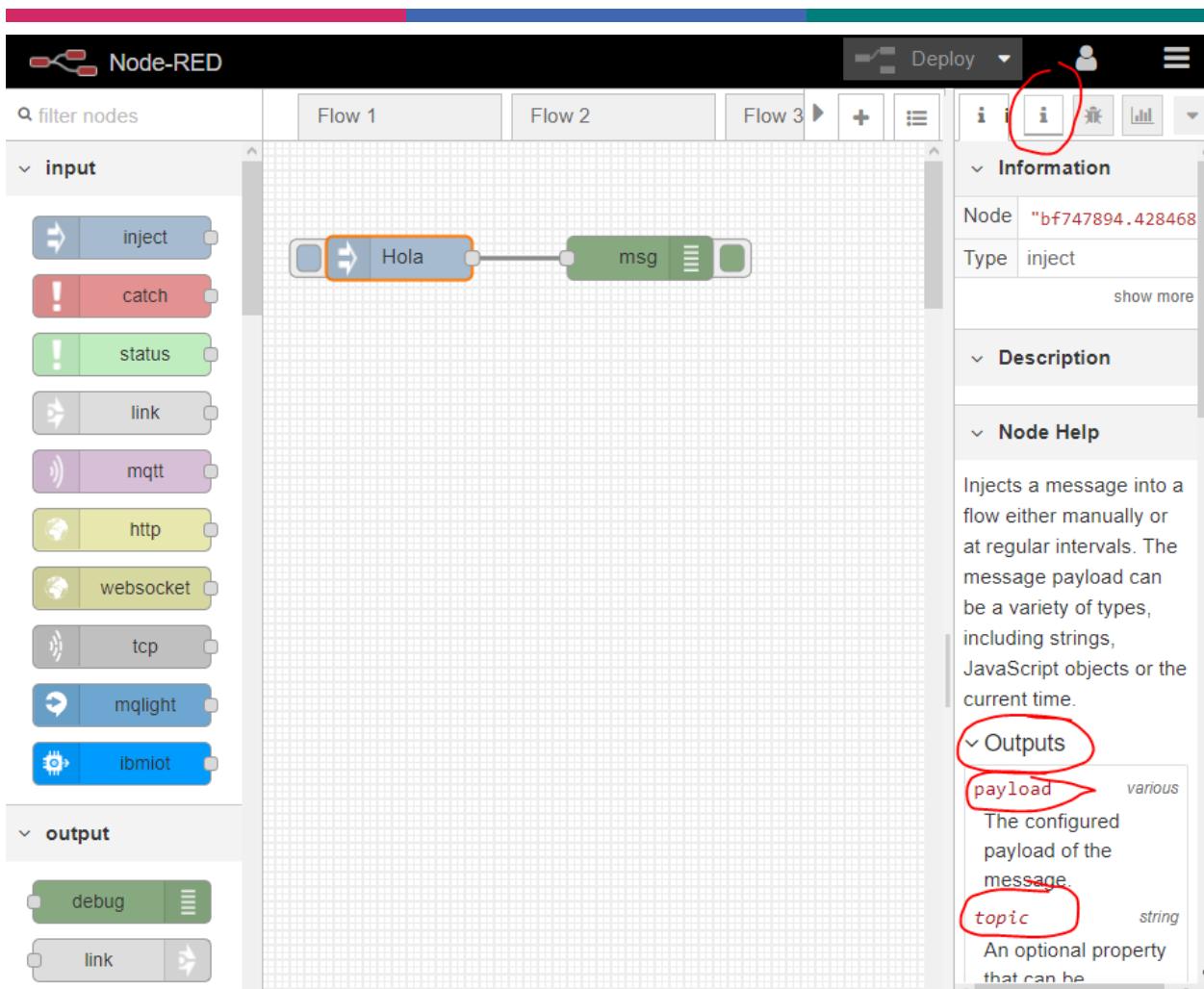


Debug nos mostrará el msg que recibe en la ventana debug para ver el contenido de sus llaves y valores..



Estos dos nodos conectados harán que debug muestre el contenido que inject está enviando. El resultado que se muestra en la ventana debug es precisamente el objeto JSON llamado msg y su contenido. Tal como puede apreciarse, el resultado son tres llaves: “_msgid” es el identificador del mensaje, “topic” y “payload” son llaves que llevan información que puede serles de utilidad a otros nodos.

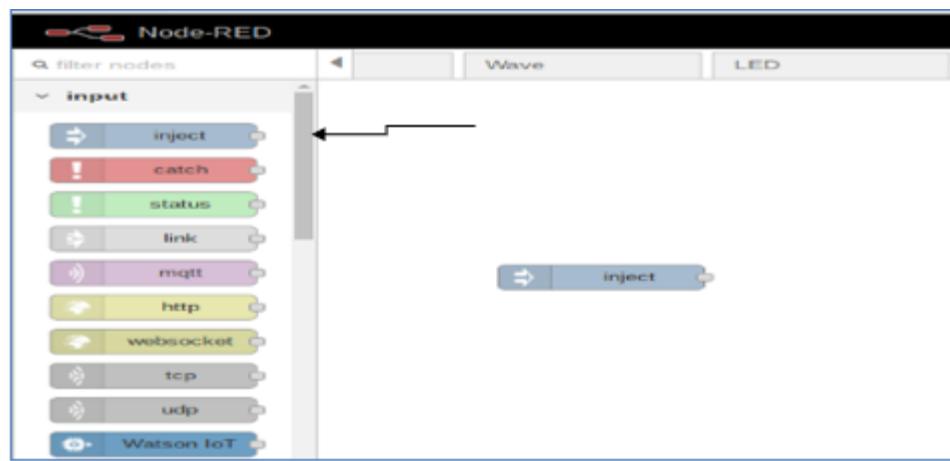
La llave “Payload” es de principal importancia ya que es convención general que la información más importante debe estar contenida ahí, por lo que la mayoría de los nodos leen o alojan sus resultados en esta llave. Si bien esto es una convención, no todos los nodos cumplen con esta regla, por lo que es nuestra tarea manipular las llaves para colocar en payload o en otras llaves la información en caso de que el nodo lo requiera. Para conocer las llaves que cada nodo utiliza como entrada y como salida podemos dar click al nodo de nuestro interés y luego seleccionar la ventana de información en la esquina superior derecha.



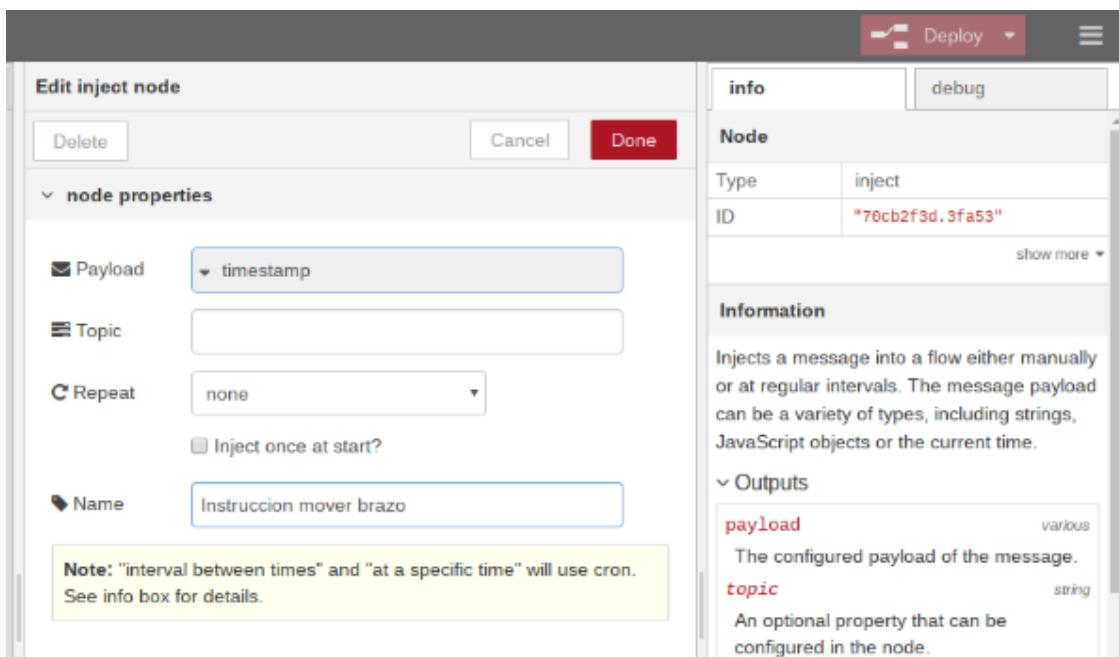
Una vez comprendido el concepto vamos a crear las dos actividades más sencillas propias de TJBot: Mover el brazo y encender su led. Estas actividades no requieren conectividad a internet ni hacen uso de los servicios de Watson para funcionar, simplemente demuestran lo fácil que es conectar elementos a los pines de la Raspberry PI para hacerlos funcionar desde Node-RED.

TJBot, mueve tu brazo

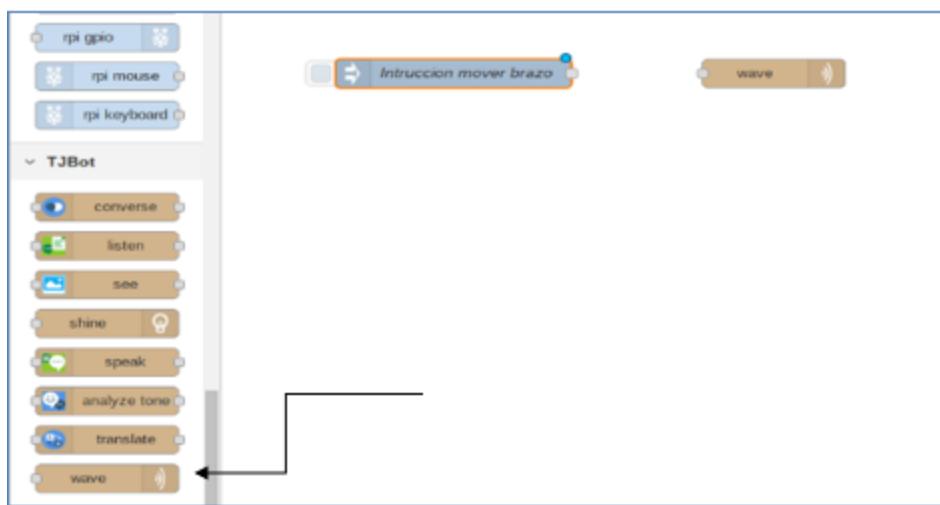
1. En el editor de Node-RED observen que en la **Paleta de Nodos** hay una sección llamada "**Input**". Seleccionen el nodo **inject** y arrástrenlo al **Espacio de Trabajo**.



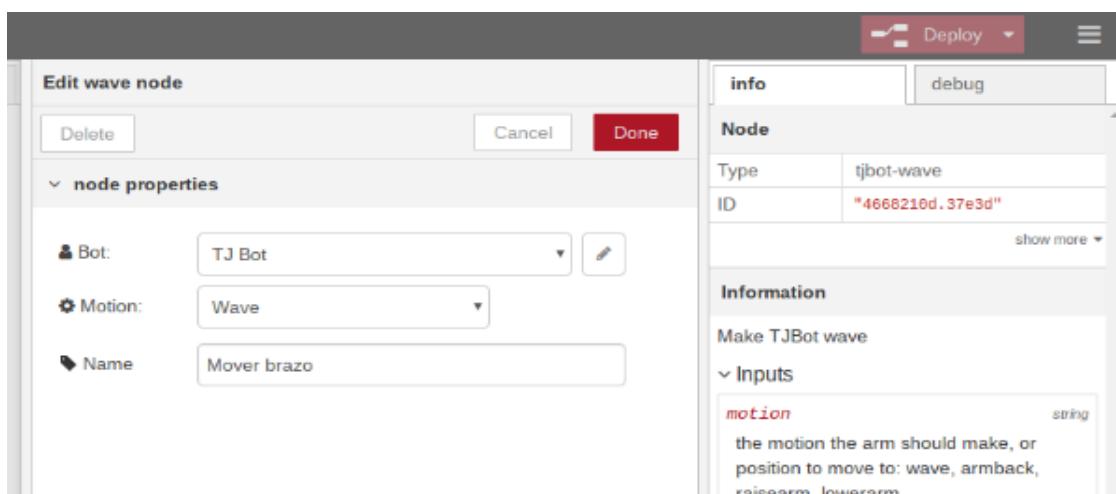
- Den doble click con el mouse al nodo **inject** que acaban de poner en el espacio de trabajo para editarlo. En la sección llamada “**Name**” escriban: **Instrucción mover brazo**. Posteriormente, en el mismo editor presionen el botón **Done**.



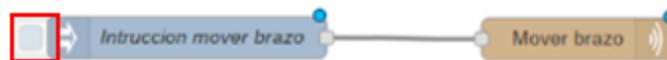
- En la **Paleta de Nodos** identifiquen la sección llamada “**TJBot**”. Selecionen el nodo **wave** de esta sección y arrástrenlo al **Espacio de Trabajo**.



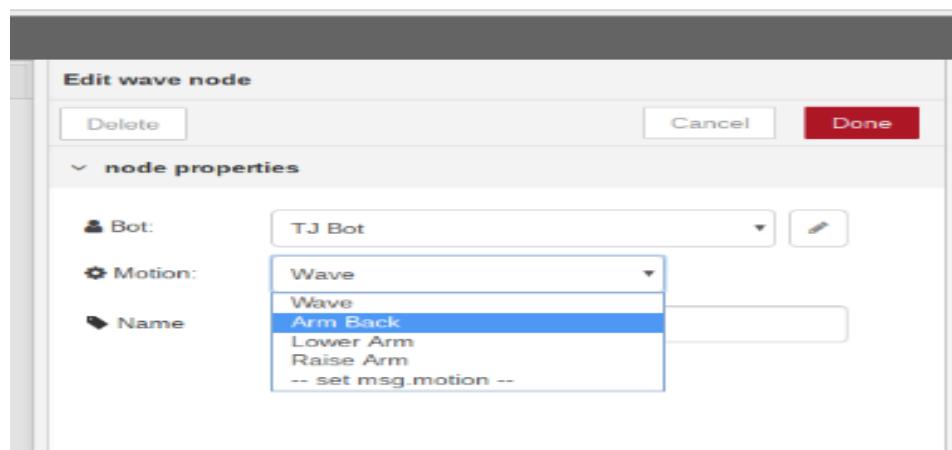
- Den doble click con el mouse al nodo **wave** para editarlo. En la sección llamada **“Name”** escriban: **Mover brazo**. Posteriormente, en el mismo editor presionen el botón **Done**.



- Coloquen el puntero del mouse en el extremo derecho del nodo **Instrucción mover brazo** y arrástralos para crear una línea que lo une con el nodo **Mover brazo**. A esto se le conoce como un **flujo (flow)**.
- Den click al **botón de Despliegue (Deploy)** localizado en la parte superior derecha del editor.
- Den click al cuadrado localizado en el extremo izquierdo del nodo **Instrucción mover brazo** para indicarle al flujo que se ejecute. Observen a **TJBot**.

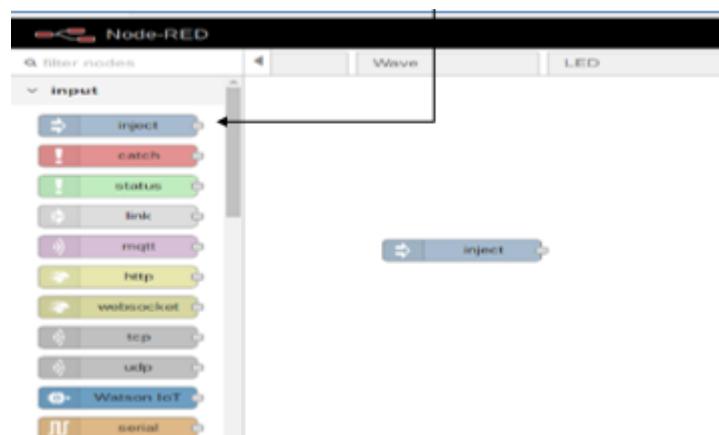


- Den doble click nuevamente al nodo **Mover brazo** para editarlo. Den click a la flecha en la sección de **Movimiento (Motion)** para desplegar las opciones. Observen que se pueden realizar varios movimientos (*Wave*, *Arm back*, *Lower Arm* y *Raise Arm*). Exploren cambiando todos los movimientos y registren en la tabla final lo que observaron en **TJBot**.

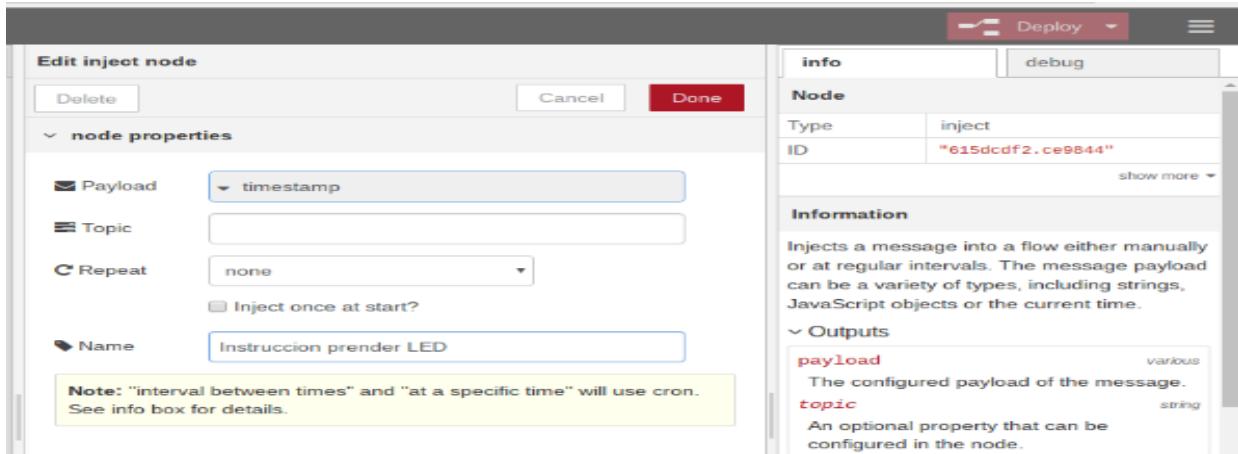


TJBot, enciende tu led

- En el editor de Node-RED observen que en la **Paleta de Nodos** hay una sección llamada "**Input**". En esa sección se encuentran diferentes nodos que contienen instrucciones para **TJBot**. Seleccíonan el nodo **Inject** y arrástrenlo al **Espacio de Trabajo**.



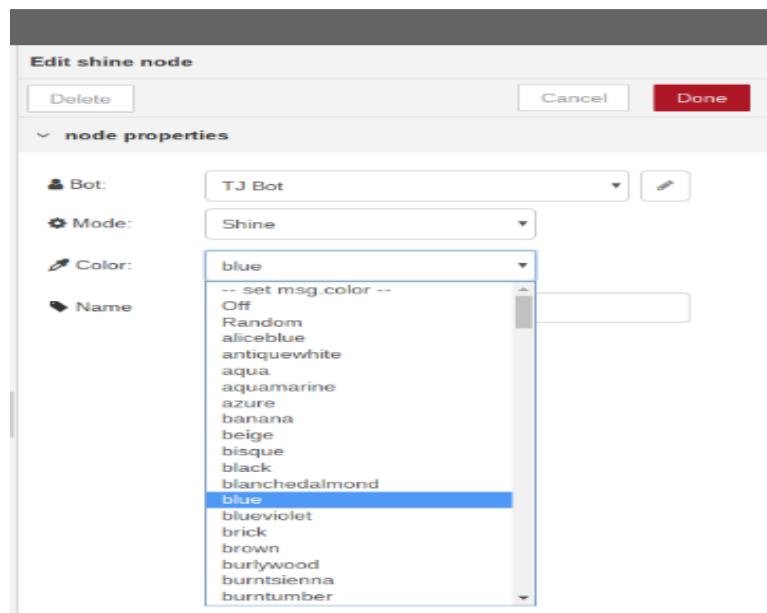
2. Den doble click con el mouse al nodo **Inject** para editarlo. En la sección llamada **“Name”** escriban: **Instrucción prender LED**. Posteriormente, en el mismo editor presionen el botón **Done**.



3. En la **Paleta de Nodos** identifiquen la sección llamada **“TJBot”**. Seleccionen el nodo **shine** de esta sección y arrástrenlo al **Espacio de Trabajo**.



4. Den doble click con el mouse al nodo **shine** para editarlo. En la sección llamada **“Name”** escriban: **Prender LED**. Den click a la flecha en la sección **Modo (Mode)** para desplegar las opciones. Observen que hay dos modos en los que se puede prender el LED (**Shine** y **Pulse**). Selecionen **“Shine”**. Posteriormente, den click en la opción **“Color”** para desplegar los colores. Selecionen el color azul (**blue**), aunque te recomendamos también la opción **Random**. Finalmente, en el mismo editor presionen el botón **Done**.



5. Coloca el puntero del mouse en el extremo derecho del nodo **Instrucción prender LED** y arrástralos para crear una línea que lo une con el nodo **Prender LED**. A esto se le conoce como un **flujo (flow)**.

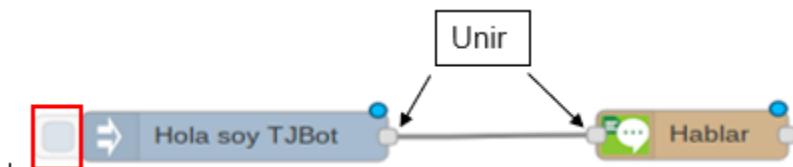


6. Den click al **botón de Despliegue (Deploy)** localizado en la parte superior derecha del editor.
7. Den click al cuadrado localizado en el extremo izquierdo del nodo **Instrucción prender LED** para indicarle al flujo que se ejecute. Observen a **TJBot**.
8. Den doble click nuevamente al nodo **Prender LED** para editarlo. Den click a la flecha en la sección “**Mode**” para desplegar las opciones. Selecciónen ahora **Pulse**, y en Duration, escriba el número 1, el cual indicará una duración de un segundo. Nuevamente den click al botón **Done**, después al **botón de despliegue (Deploy)** en el editor y vuelvan a dar click en el cuadrado al lado izquierdo del nodo **Instrucción prender LED** para ejecutar la instrucción. Observen ahora el comportamiento de **TJBot**.

TJBot... ¡Habla!

Es momento de utilizar uno de los servicios de Watson: Text to Speech. Este servicio tomará el texto y sintetizará una voz humana, asegúrate de haber agregado la llave de API correspondiente en la configuración del bot.

1. En el editor de Node-RED observe que en la **Paleta de Nodos** hay una sección llamada “**Input**”. Seleccione el nodo **inject** y arrástrello al **Espacio de Trabajo**.
2. Dé doble click con el mouse al nodo **inject** para editarlo. En la sección llamada “**Payload**” dé click a la flecha para ver las opciones y seleccionen **string**. Una vez seleccionado, escriba en ese espacio: **Hola, soy TJBot**. Finalmente, en el mismo editor presione el botón **Done**.
3. En la **Paleta de Nodos**, identifique la sección llamada “**TJBot**”. Seleccione el nodo **Speak** de esta sección y arrástrello al **Espacio de Trabajo**. Dé doble click con el mouse al nodo **speak** para editarlo.
4. En la sección llamada “**Mode**” selecciona el modo **Speak**, y en la sección llamada “**Name**” escriba: **Hablar**, y después en el mismo editor de click en **Done**.
5. De regreso al espacio de trabajo en Node Red, una los dos nodos utilizando el puntero del mouse. De click al círculo en el nodo de “Hola soy TJBot”, sin separar el dedo del mouse alargue el puntero hasta el círculo izquierdo en el nodo de “Hablar”. A esto se le conoce como un **flujo (flow)**. Use la siguiente imagen como referencia:



6. Dé click al **botón de Despliegue (Deploy)** localizado en la parte superior derecha del editor.
7. Dé click al cuadrado localizado en el extremo izquierdo del nodo **Hola soy TJBot** para indicarle al flujo que se ejecute. Escuchen a **TJBot**.

TJBot... ¡Escucha!

Esta actividad hace uso del servicio *Speech to Text* de Watson por medio del nodo listen. Una particularidad en el nodo listen es que para iniciar o detener su escucha hace uso de la llave “mode”, donde le deberemos indicar si debe iniciar su escucha (start) o detenerla (stop). Para colocar este valor en esta nueva llave haremos uso del nodo “change”, el cual es capaz de crear nuevas llaves o modificar las existentes con el contenido que nosotros indiquemos.

1. De la **Paleta de Nodos**, seleccionen el nodo **inject** y arrástrenlo al **Espacio de Trabajo**.
2. Den doble click con el mouse al nodo **inject** para editarlo. En la sección llamada “**Payload**” den click a la flecha para ver las opciones y seleccionen **string**. Una vez seleccionado escriban en ese espacio: **start**. Posteriormente, en la sección llamada “**Name**” escriban: **Comenzar a escuchar**. Finalmente, en el mismo editor presionen el botón **Done**.

Edit inject node

Edit inject node

node properties

node properties

Payload: timestamp

Payload: start

Topic

Topic

Repeat: none

Repeat: none

Name: Comenzar a escuchar

Note: "interval between times" and "at a specific time" will use cron. See info box for details.

3. Seleccionen otro nodo **inject** de la **Paleta de Nodos** y arrástrenlo al **Espacio de Trabajo**. Den doble click con el mouse al nuevo nodo **inject** para editarlo. En la sección llamada “**Payload**” den click a la flecha para ver las opciones y seleccionen **string**. Una vez seleccionado escriban en ese espacio: **stop**. Posteriormente, en la sección llamada “**Name**” escriban: **Parar de escuchar**. Finalmente, en el mismo editor presionen el botón **Done**.

Edit inject node

Edit inject node

node properties

node properties

Payload: stop

Payload: stop

Topic

Topic

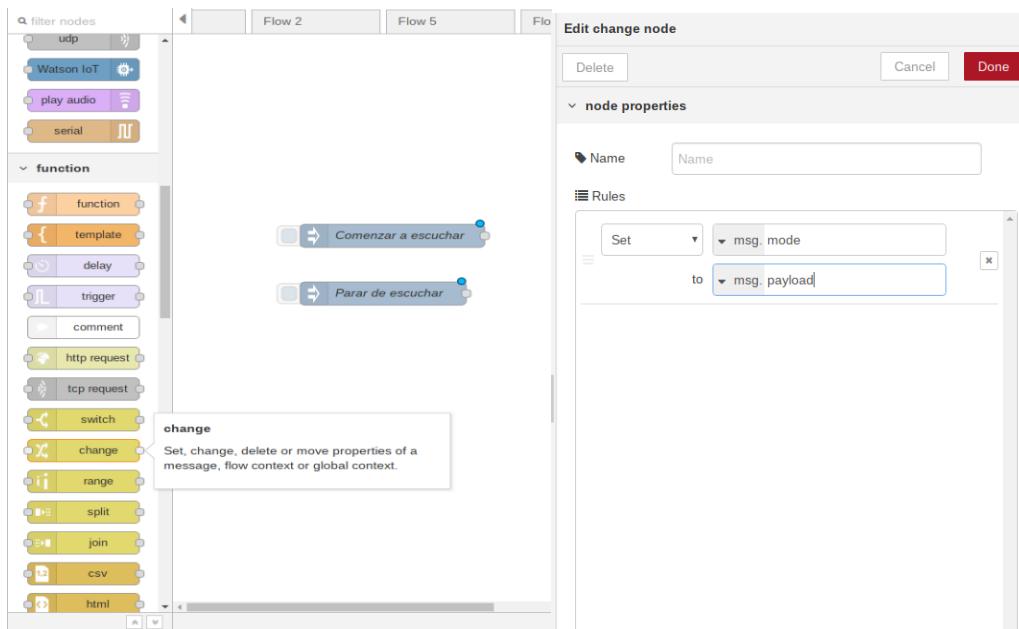
Repeat: none

Repeat: none

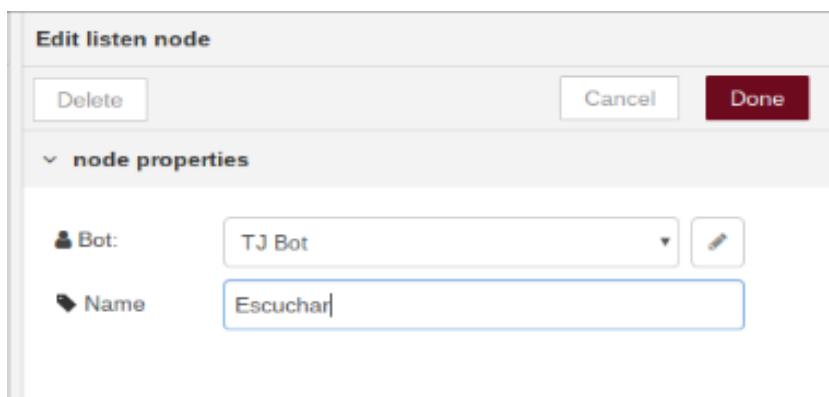
Name: Parar de escuchar

Note: "interval between times" and "at a specific time" will use cron. See info box for details.

4. En la **Paleta de Nodos** identifiquen la sección llamada “**function**”. Selecionen el nodo **change** de esta sección y arrástrenlo al **Espacio de Trabajo**. Den doble click a este nodo para editarlo. Observen los dos recuadros en la sección “**Set**”, con las flechas que están ahí desplieguen las opciones, elijan la opción “**msg**” en ambos recuadros, en el recuadro de arriba escriban “**mode**” y en el de abajo “**payload**”. Usa la imagen siguiente como referencia:



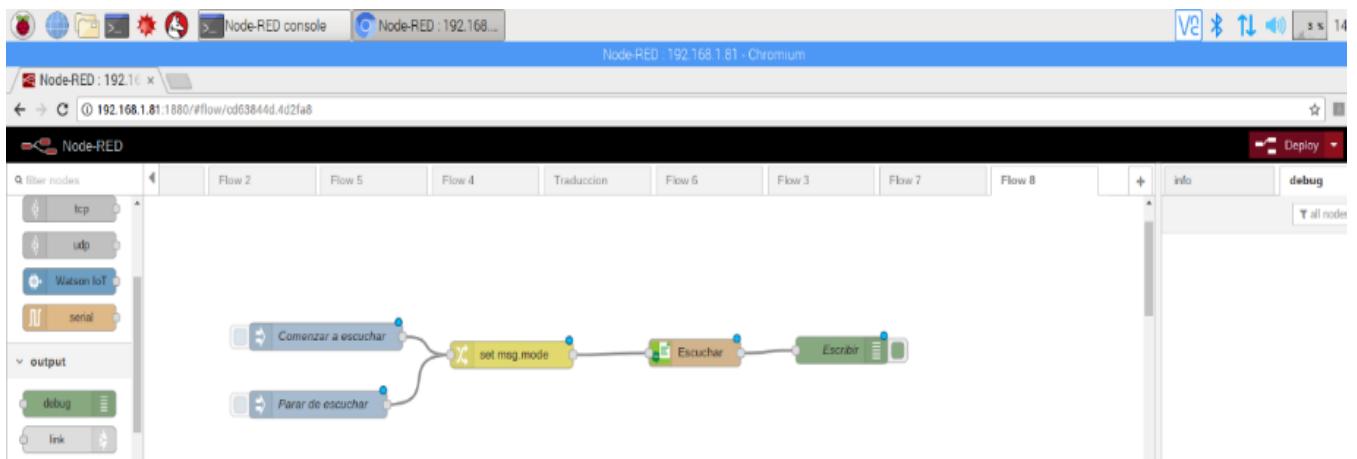
5. En la **Paleta de Nodos** identifiquen la sección llamada “**TJBot**”. Selecionen el nodo **Listen (escuchar)** de esta sección y arrástrenlo al **Espacio de Trabajo**. Den doble click con el mouse al nodo **listen** para editarlo. En la sección llamada “**Name**” escriban: **Escuchar**. Posteriormente, en el mismo editor presionen el botón **Done**.



6. En la **Paleta de Nodos** identifiquen la sección llamada “**Output**”. Seleccionen el nodo **debug (depurar)** de esta sección y arrástrello al **Espacio de Trabajo**. Den doble click con el mouse a este nodo para editarlo. En la sección llamada “**Name**” escriban: **Escribir**. Posteriormente, en el mismo editor presionen el botón **Done**.
7. Identifiquen la ventana de depuración “debug” del editor y den click para abrirla.



8. Unan los nodos como se muestra en la siguiente imagen:

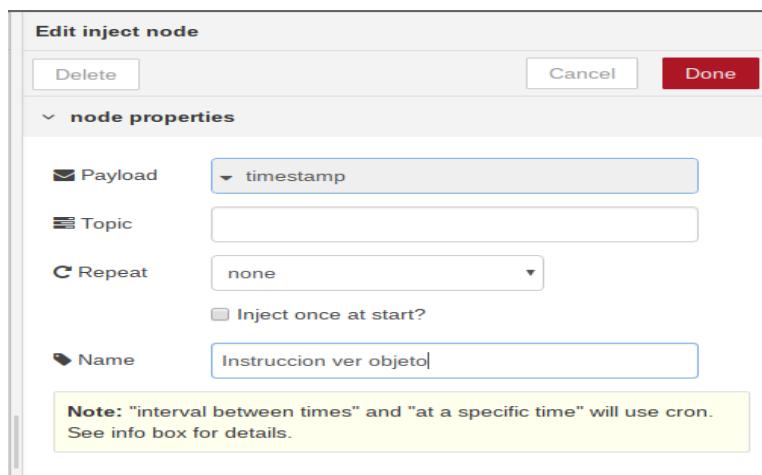


9. Den click al botón de Despliegue (**Deploy**) localizado en la parte superior derecha del editor.
10. Den click al cuadrado localizado en el extremo izquierdo del nodo **Comenzar a escuchar** para indicarle al flujo que se ejecute. Acérquense a **TJBot** y díganle una palabra. Observen la ventana de depuración. ¿Qué sucedió? Intenten decirle diferentes palabras y observen el resultado. Cuando quieran que se detenga de escuchar, den click al cuadrado localizado en el extremo izquierdo del nodo **Parar de escuchar**.

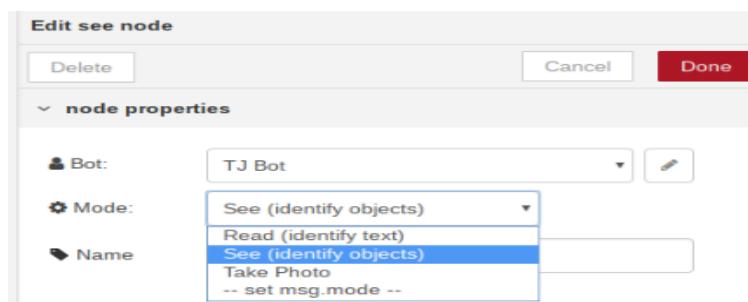
TJBot... ¡Observa!

Es el servicio de Visual Recognition el que le da la capacidad de reconocer objetos a TJBot. Si bien este es uno de los servicios que más variantes puede tener, en este momento mostraremos el uso más básico del nodo, reconocimiento de objetos en general.

1. En la **Paleta de Nodos**, seleccionen el nodo **inject** y arrástrenlo al **Espacio de Trabajo**. Ya en el **Espacio de trabajo**, den doble click con el mouse a dicho nodo para editarlo. En la sección llamada “**Name**” escriban: **Instrucción ver objeto**. Posteriormente, en el mismo editor presionen el botón **Done**.



2. En la **Paleta de Nodos** identifiquen la sección llamada “**TJBot**”. Selecionen el nodo **see (ver)** de esta sección y arrástrenlo al **Espacio de Trabajo**. Den doble click con el mouse al nodo **see** para editarlo. En la sección llamada “**Mode**”, desplieguen con la flecha las opciones y seleccionen: **See (identify objects)**. Posteriormente, en la sección llamada “**Name**” escriban: **Identificar objeto**. Finalmente, en el mismo editor presionen el botón **Done**.



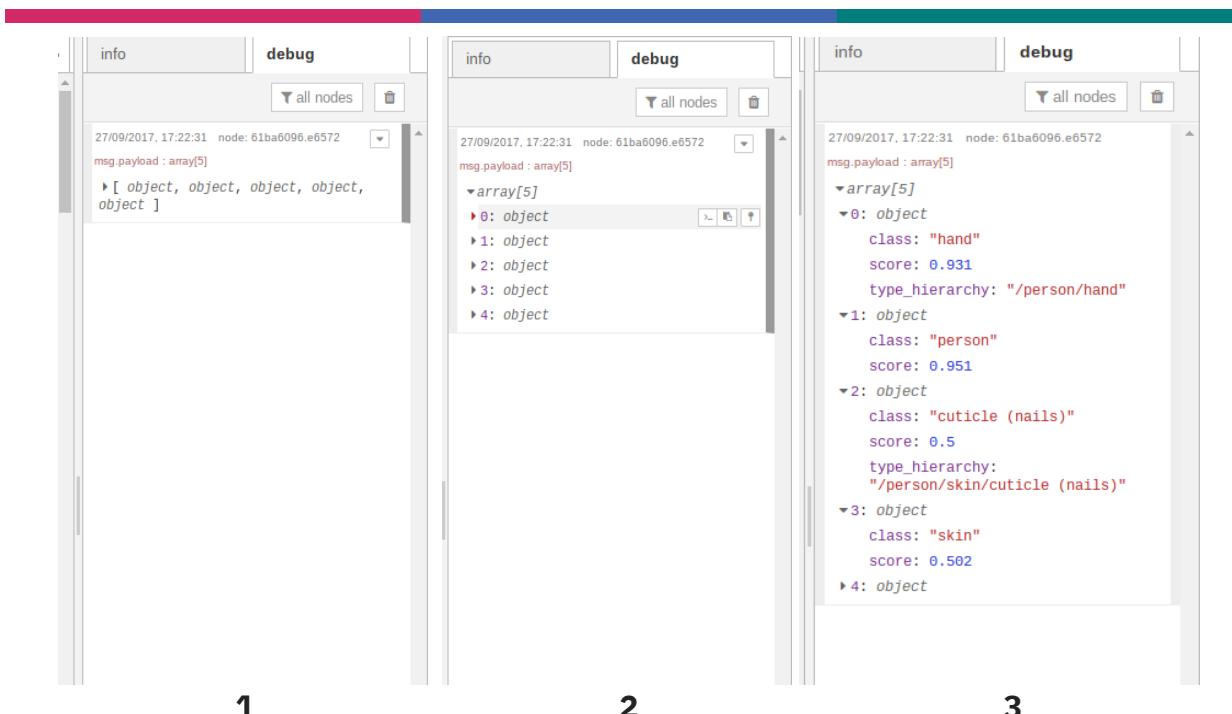
3. En la **Paleta de Nodos** identifiquen la sección llamada “**Output**”. Seleccionen el nodo **debug (depurar)** de esta sección y arrástrenlo al **Espacio de Trabajo**.
4. Identifiquen la ventana de depuración del editor y den click para abrirla.



5. Unan los nodos como se muestra en la siguiente imagen:



6. Den click al **botón de Despliegue (Deploy)** localizado en la parte superior derecha del editor.
7. Uno de ustedes ponga la palma de su mano a aproximadamente 20 cm del ojo de TJBot que tiene la cámara. Den click al cuadrado localizado en el extremo izquierdo del nodo **Instrucción ver objeto** para indicarle al flujo que se ejecute.
8. Observen que apareció información en la ventana de depuración. Den click a las puntas de flecha que vayan apareciendo hasta que lleguen a algo similar de lo observado en la imagen 3



1

2

3

9. Junto a la palabra **Class** encontrarán los resultados que les entrega TJBot con respecto a palabras que describen al objeto que observó. **¿Los resultados corresponden a palabras que se podrían utilizar para describir correctamente la mano que vio TJBot?** En caso negativo, repitan el procedimiento desde el paso 7. Probablemente tengan que acercar o alejar más la mano o asegurarse que el cartón de TJBot no esté tapando la cámara.

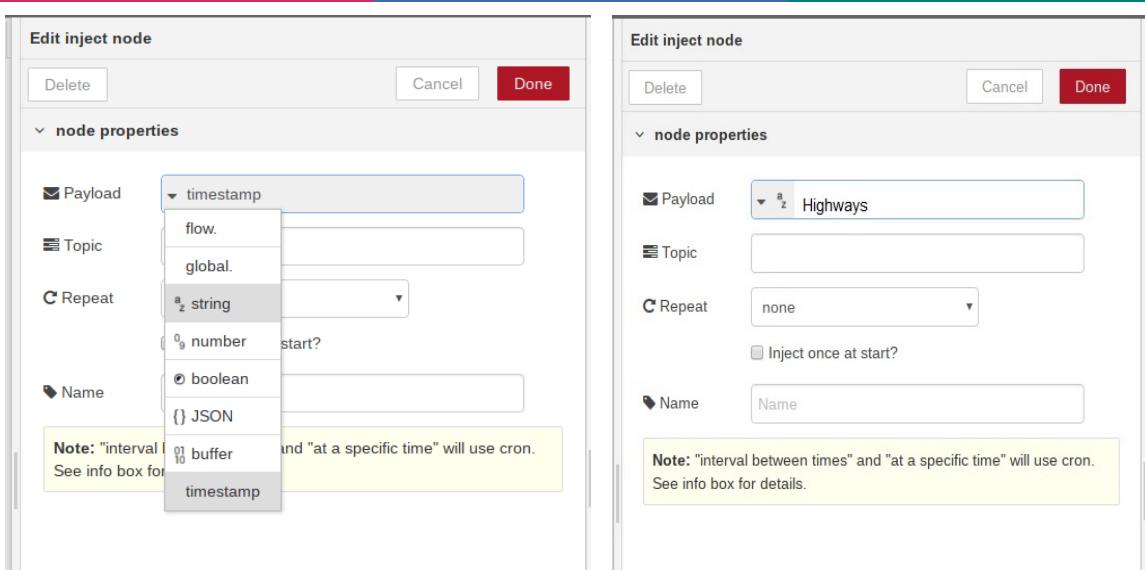
Nota:

Los resultados que entrega TJBot están en idioma inglés.

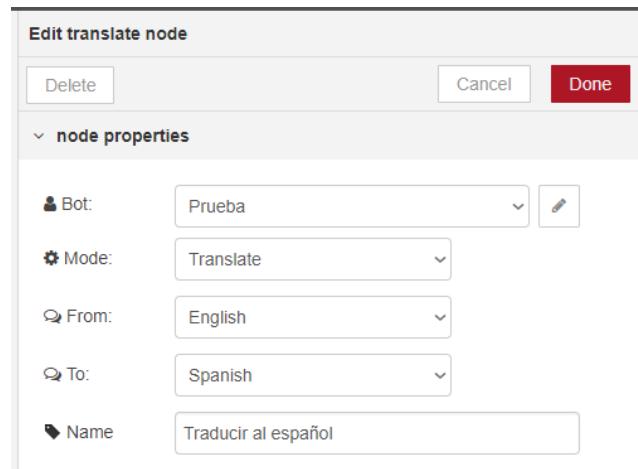
TJBot... ¡Traduce!

La traducción que es capaz de realizar TJBot es muy variada pues se dispone de más de 20 lenguajes, con la capacidad de además identificar idiomas, esto se logra gracias al servicio Language Translation.

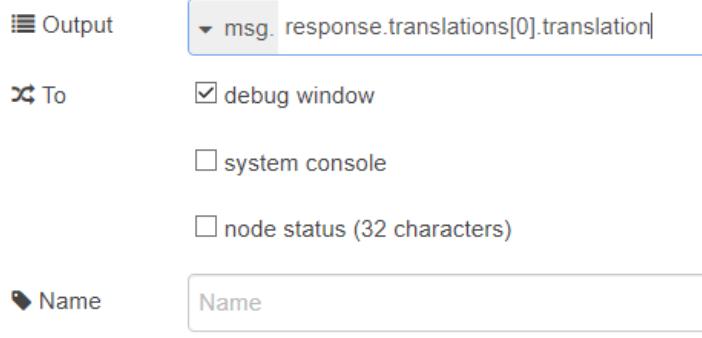
1. De la **Paleta de Nodos**, seleccionen el nodo **inject** que se encuentra en la sección “**Input**” y arrástrenlo al **Espacio de Trabajo**.
2. Den doble click con el mouse al nodo **inject** para editarlo. En la sección llamada “**Payload**” den click a la flecha para ver las opciones y seleccionen **string**. Una vez seleccionado escriban en ese espacio: **Highways**. Den click al botón Done.



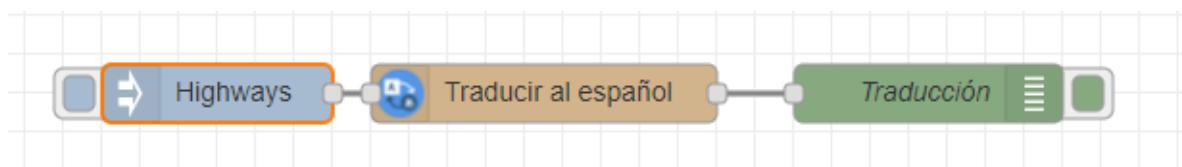
3. De la **Paleta de Nodos**, seleccionen el nodo ***language translator*** (**Traductor de lenguaje**) que se encuentra en la sección “**TJBot**” y arrástrenlo al **Espacio de Trabajo**. Den doble click a este nodo para editarlo. En la opción **From** selecciones el idioma “**English**” el idioma origen, para decirle a cual idioma queremos que traduzca desplegamos las opciones de la sección **To** y elegiremos “**Spanish**” (**español**). Posteriormente, escriban en la sección “**Name**”: **Traducir al español**. Finalmente, en el mismo editor presionen el botón **Done**.



4. En la **Paleta de Nodos** identifiquen la sección llamada “**Output**”. Seleccionen el nodo ***debug (depurar)*** de esta sección y arrástrenlo al **Espacio de Trabajo**. En output colocar “*response.translations[0].translation*”, por nombre, escriban “**Traducción**”. Finalmente, en el mismo editor presionen el botón **Done**.



5. Unan los nodos como se muestra en la siguiente imagen:



6. Identifiquen la ventana de depuración del editor y den click para abrirla.



7. Den click al **botón de Despliegue (Deploy)** localizado en la parte superior derecha del editor.
8. Den click al cuadrado localizado en el extremo izquierdo del nodo **Highways** para indicarle al flujo que se ejecute. Observen la ventana de depuración. **¿Qué sucedió?**

Actividades avanzadas

La verdadera fortaleza de la programación con TJBot se encuentra en la facilidad de integrar diferentes tecnologías con una facilidad digna de un experto programador. Las actividades avanzadas son solo algunos ejemplos de cómo podemos fusionar algunas de estas tecnologías para generar aplicaciones novedosas y de mucha utilidad.

Comenzar a trabajar con diferentes tecnologías en una misma aplicación nos obliga a conocer el concepto de una API. 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.

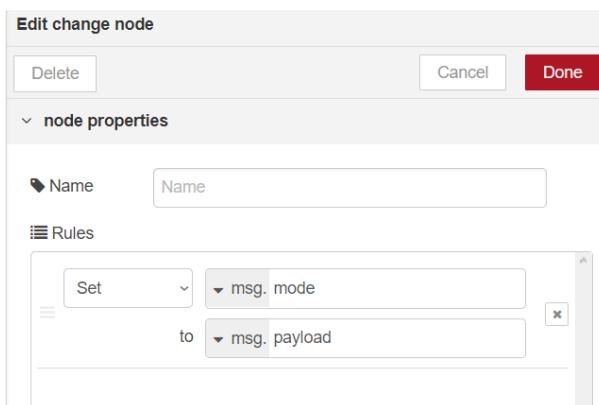
Existen todos tipos de APIs, desde las que te permiten consultar las noticias, obtener música, información sobre el clima, entre otros. Para poder usar esos servicios necesitamos una llave de acceso, la cual puede ser un Token (Código de acceso), o un usuario y contraseña. Esto se puede obtener en la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

Como convención general las APIs envían y reciben mensajes en formato JSON para cualquiera de sus funcionalidades: Autenticación, recepción de información o envío de resultados.

Esto sugiere nuevamente que nosotros en nuestra interfaz Node-RED podamos manipular nuestros JSON de tal manera que podamos transmitir y recibir la información proveniente de estas APIs, por lo que exploraremos tres nodos más que nos resultarán útiles durante las siguientes actividades.

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.



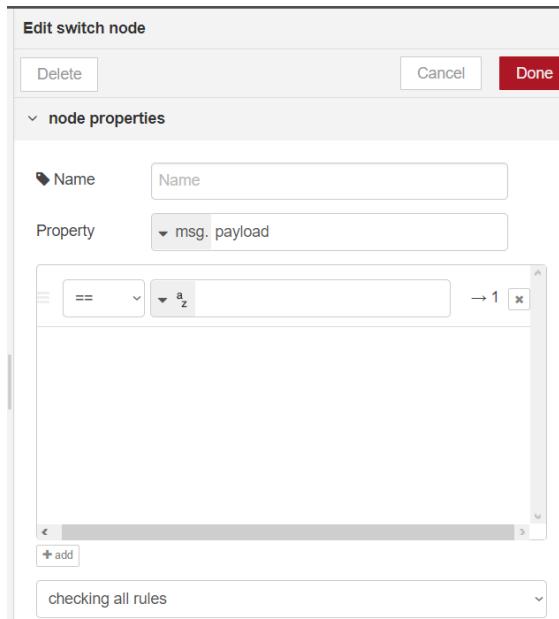
En el ejemplo anterior, tenemos un menú desplegable en Set, que copiará el contenido del primer campo al segundo. Además, se cuenta con las opciones de Change (cambiar), Delete (eliminar), Move (mover), para manipular la información del JSON que cruce por el nodo.

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.

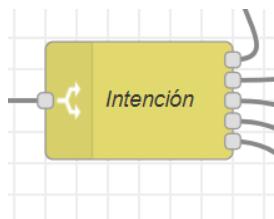


En el campo Property, se indica la entrada que el nodo comparará. Y en el menú desplegable, se indica el tipo de comparación a realizar. A continuación, se presentan las comparaciones más comúnmente utilizadas:

Símbolo	Significado
<code>==</code>	Igual a
<code>!=</code>	Diferente a
<code><</code>	Menor a
<code>></code>	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

Instalación de nuevos nodos

Al ser de código abierto, programadores han creado una diversidad de nodos para distintos propósitos que pueden ser agregados a la interfaz: Nodos para redes sociales, servicios de streaming, con conexión a APIs externas para obtener información de todo tipo, etcétera.

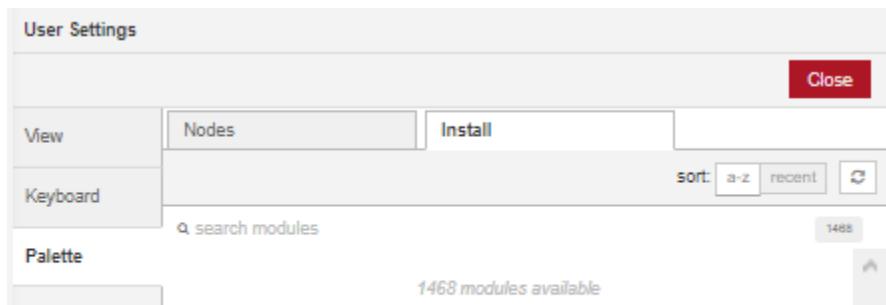
El listado de nodos disponibles para instalar se encuentran en la siguiente liga: <https://flows.nodered.org/>

Para realizar la instalación de estos nodos:

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



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

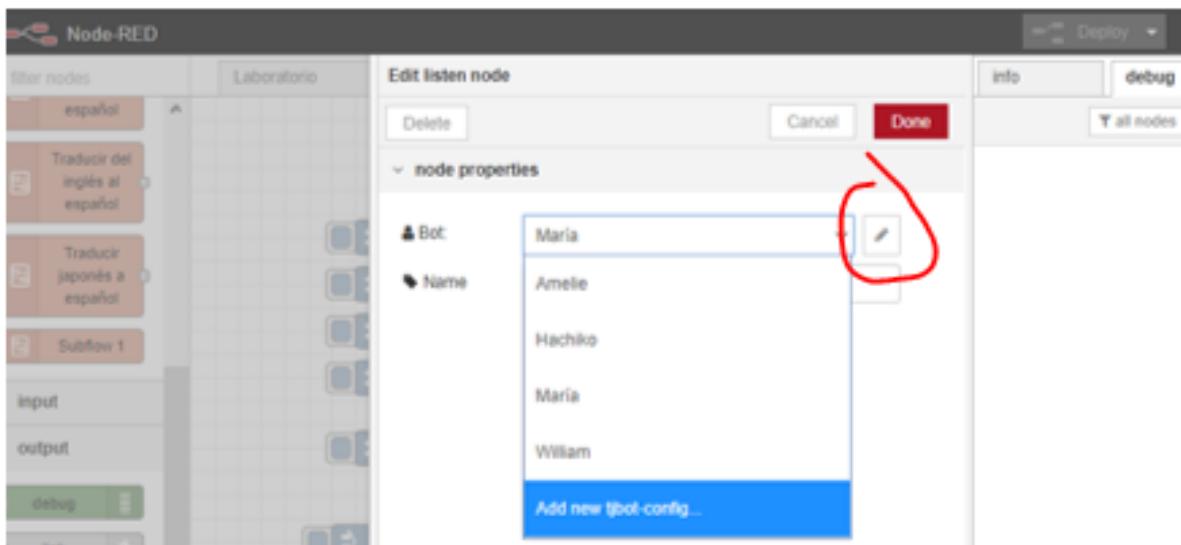


3. En el buscador **search modules** busque la palabra clave o el nombre de la paquetería específica que desea instalar.
4. Espere a que concluya la instalación, habrá un mensaje de confirmación y cuando esto suceda verá los nuevos nodos en la paleta de nodos del lado izquierdo.

Instrucciones para que TJBot escuche y traduzca entre idiomas

En esta actividad integraremos tres de las actividades básicas que pudimos programar anteriormente para que TJBot pueda escuchar un idioma, traducirlo y hablarlo en otro idioma.

Para esto 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. Al ser este bot nuevo aún no cuenta con credenciales, agregue solamente las credenciales **Speech to text** y **Text to speech**, puede usar las mismas con las que creó a su primer Bot, y en la sección **Name** escriba TJBot inglés. Da click en Add, y luego en done.

Edit listen node > Edit tjbot-config node

Delete Cancel Update

Gender: Male

Speak: English (British dialect)

Listen: UK English

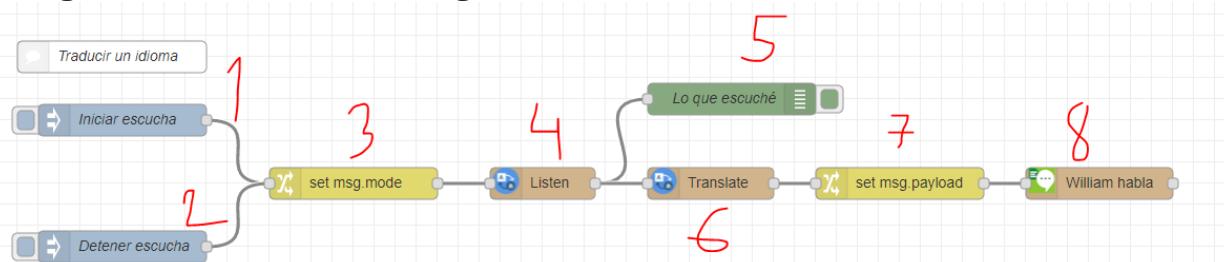
Has

- Servo
- LED
- Camera
- Microphone
- Speaker

Name: William

A partir de este punto, tendremos dos Bots configurados: Uno habla y escucha español y el otro habla y escucha inglés.

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:



1. Del apartado input, nodo inject. **Payload: string:** start. **Name:** Iniciar escucha.
2. Del apartado input, nodo inject. **Payload: string:** stop. **Name:** Detener escucha.
3. Del apartado function, nodo change.



4. Del apartado TJBot, nodo listen. **Bot:** María.
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.
6. Del apartado TJBot, nodo translate. **Mode:** Translate. **From:** Spanish. **To:** English. **Name:** español a inglés.
7. Del apartado function, nodo change.



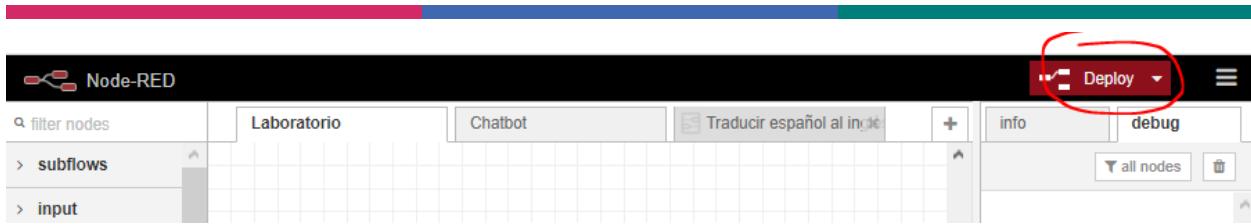
8. Del apartado TJBot, nodo speak. **Bot:** William, **Mode:** speak, **Name:** William habla.

Descripción del flujo.

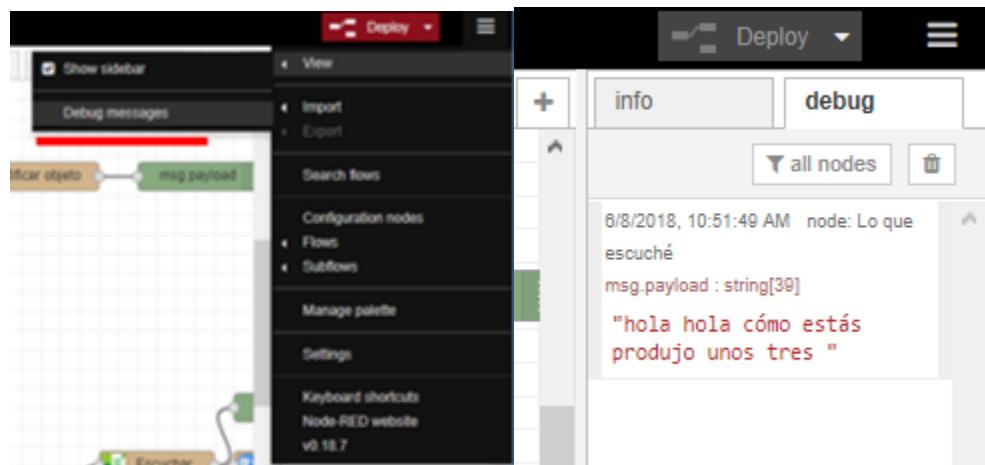
Los nodos **1, 2, 3 y 4** nos permitirán iniciar y detener el escucha del robot. Podemos ver que el nodo **3: Change** está cargando la instrucción **stop y start** que sale de la llave mode, a la llave payload, que es la que el nodo **4: Listen** es la que lee por default. El nodo **5 Lo que escuché** 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, 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:



¿Qué más puedo hacer?

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

1. En el nodo **Listen**, cambiar a María por William.
2. En el nodo **Español a inglés**, cambiar **From** a **English**, y **To** 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 **From** y el **To** 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.

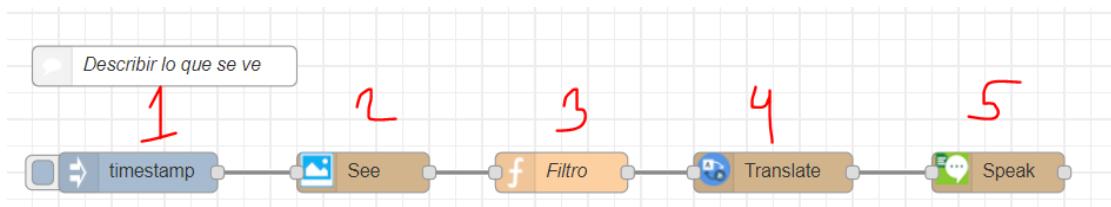
Hay algunas APIs que funcionan mejor cuando usas el idioma inglés para interactuar con ellas, como por ejemplo la funcionalidad de análisis de sentimientos. Puedes aprovechar la traducción de este flujo para interactuar con estas funciones.

Instrucciones para que TJBot describa 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:



1. Del apartado input, nodo inject. **Payload**: timestamp, **Name**: Ver objeto.
2. Del apartado TJBot, nodo see. **Mode**: See (identify objects), **Name**: Identificar objeto.
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.

4. Del apartado TJBot, nodo translate. **Mode**: Translate. **From**: Spanish. **To**: English. **Name**: español a inglés.
5. Del apartado TJBot, nodo Speak. **Bot**: María.

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.

Instrucciones para que TJBot twittee 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 user settings interface. At the top, there is a navigation bar with links for Weather, Maps, API, Price, Partners, Stations, Widgets, News, and About. Below the navigation bar, the page title is "My Home". On the left, there is a sidebar with links for Setup, API keys (which is circled in red), My Services, My Payments, Billing plans, Map editor, Block logs, History bulk, User Settings (highlighted in orange), Privacy Centre, and Invoice Info. The main content area shows form fields for Username, Email, and Full name, each with a corresponding redacted input field. A "Save" button is located at the bottom left of the form. In the top right corner, there is a "Logout" button.

3. Copia tu API key para utilizarlo más adelante.

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.

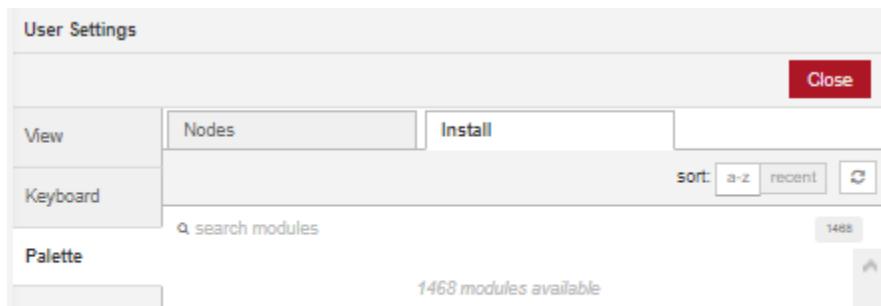
The screenshot shows a table with one row. The first column is labeled "Key" and contains a long string of characters, which is highlighted with a large red oval. The second column is labeled "Name" and has a "Default" entry. The third column contains "Create key" and "Delete" buttons. A text input field labeled "* Name" is also present.

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

4. En la esquina superior derecha, del menú desplegable selecciona **Manage palette**.

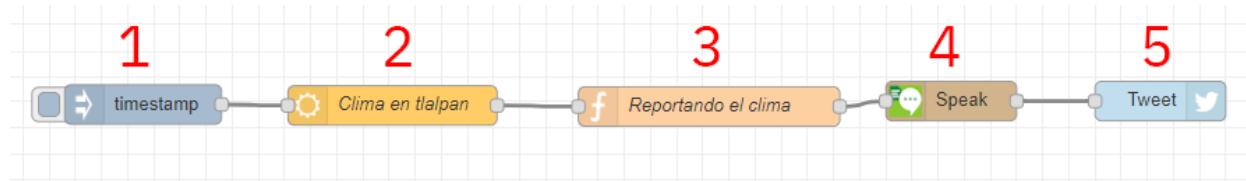


5. De las opciones a la izquierda del menú desplegado, selecciona **Palette**, y de las pestañas selecciona **Install**.



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

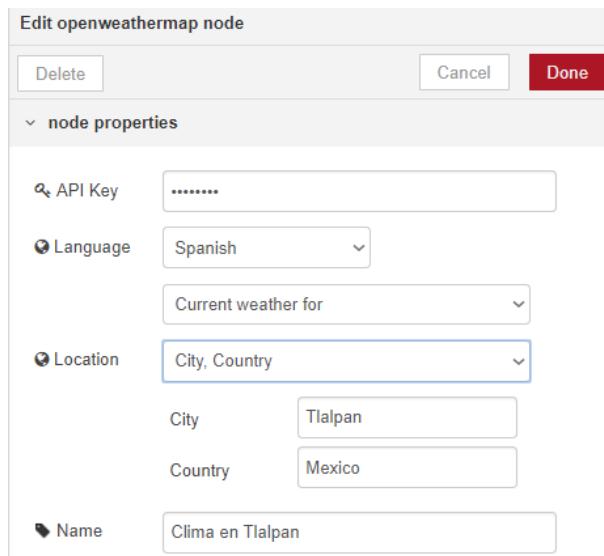
Finalmente, construiremos el flujo para llevar a cabo la actividad:



1. Del apartado input, nodo input. No será necesario configurarlo.
2. Del apartado weather, nodo openweathermap (con una entrada y una salida).

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.



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

4.

```

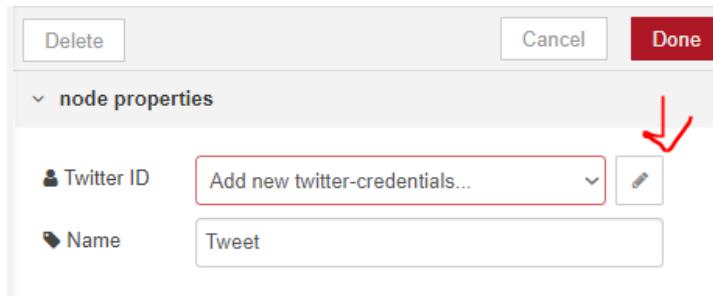
msg.payload = "El reporte del clima es, " + msg.payload.detail
+ ", tenemos una temperatura de " + msg.payload.tempc
+ "grados, y humedad máxima del " + msg.payload.humidity
+ " porcientos.";
return msg;
  
```

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.

5. Del apartado TJBot, nodo speak. **Bot:** María, **Mode:** Speak, **Name:** Reportar.

6. 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.

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

Cancel | Add

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, entonces aparecerá la siguiente advertencia donde tendremos que aplicar para una cuenta de desarrollador. Damos click al botón correspondiente y seguiremos las instrucciones que se nos pidan.

Twitter Apps

As of July 2018, you must [apply for a Twitter developer account](#) and be approved before you may create new apps. Once approved, you will be able to create new apps from developer.twitter.com.

For the near future, you can continue to manage existing apps here on apps.twitter.com. However, we will soon retire this site and consolidate all developer tools, API access, and app management within the developer portal at [developer.twitter.com](#). You will be able to access and manage existing apps through that portal when we retire this site.

[Apply for a developer account](#)

Una vez realizado el proceso y teniendo acceso de desarrollador, procederemos a crear nuestra aplicación y daremos click en el botón “Create new app”.



Twitter Apps

Llenaremos los campos que nos pide Twitter como apunta la imagen a continuación:

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.

Developer Agreement

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

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. 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”.

The screenshot shows the Twitter Application Management interface. At the top, there's a blue header bar with the Twitter logo and the text "Application Management". Below the header, the main title "Aplicación del clima" is displayed in large bold letters. Underneath the title, there are four tabs: "Details", "Settings", "Keys and Access Tokens", and "Permissions". The "Keys and Access Tokens" tab is highlighted with a red underline. In the top right corner of the main area, there's a button labeled "Test OAuth".

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

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.

This screenshot shows the "Your Access Token" section. It features a heading "Token Actions" and a prominent button labeled "Create my access token". A red circle highlights this button, indicating it is the next step to be taken.

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

Aplicación del clima

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

Consumer Key (API Key) [REDACTED]
Consumer Secret (API Secret) [REDACTED]

Access Level Read and write (modify app permissions)
Owner Jair_LizarragaV
Owner ID 142041689

Application Actions
Regenerate Consumer Key and Secret Change App Permissions

Your Access Token
This access token can be used to make API requests on your own account's behalf. Do not share it with anyone else.

Access Token [REDACTED]
Access Token Secret [REDACTED]

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

1. Create your own application at apps.twitter.com
2. From the 'Keys and Access Tokens' section, copy the Consumer Key and Consumer Secret
3. Create a new 'Access Token' and copy the Access Token and Access Token Secret

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

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

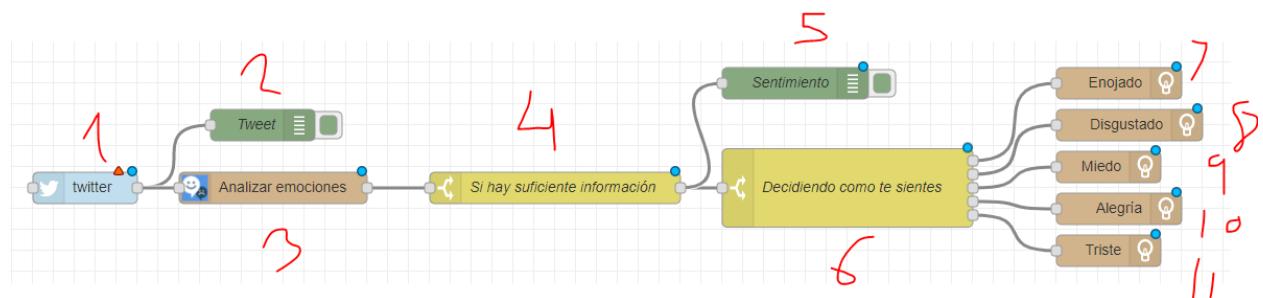
Bastará con dar click al botón azul del nodo input, para que se realice la acción. Espera unos segundos, y escucharás el reporte del clima por la bocina. Cuando termine el flujo,

visita tu perfil de Twitter, y verás que automáticamente se ha realizado una publicación en tu perfil acerca del clima.

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](#).

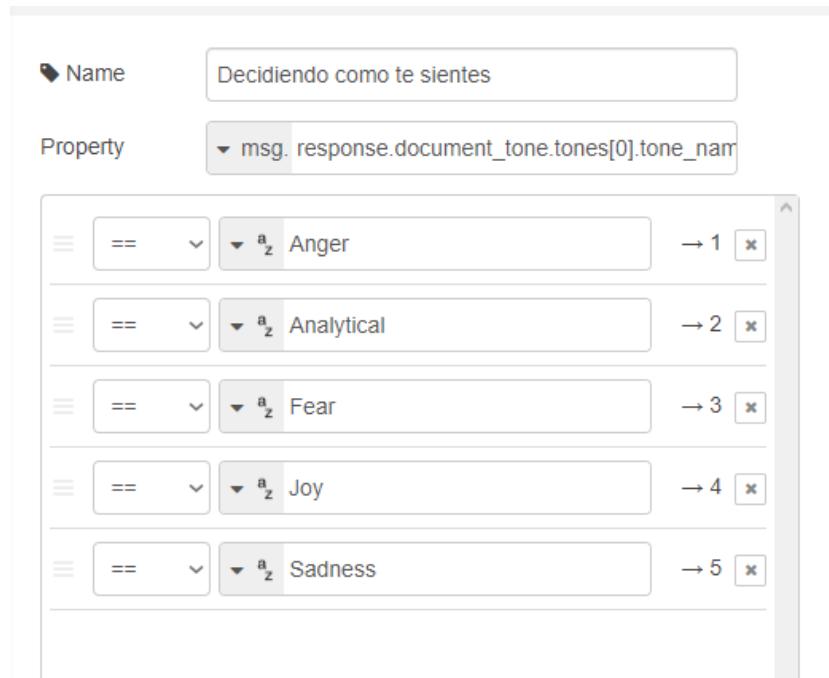
Construye el siguiente flujo, arrastrando el nodo solicitado al espacio de trabajo, uniéndolo al nodo correspondiente, y dando doble click para configurar cada nodo de acuerdo con las siguientes indicaciones:



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**.
2. Del apartado output, nodo Debug. **Name:** Tweet.
3. Del apartado de TJBot, nodo analyze tone. **Bot:** María, **Name:** Analizar emociones.
4. Del apartado function, nodo switch. **Name:** Si hay suficiente información. La configuración es la siguiente:

Name	Si hay suficiente información
Property	msg.response.document_tone.tones[0].score
<input type="button" value="is not null"/> → 1 <input type="button" value="x"/>	

5. Del apartado output, nodo Debug. **Name:** Sentimiento.
6. Del apartado function, nodo switch. Se analizarán cinco opciones, una para cada sentimiento. La configuración es la siguiente:



En property va: `msg.response.document_tone.tones[0].tone_name`.

7. Del apartado TJBot, nodo shine. Mode: Shine, Color: red, Name: Enojado.
8. Del apartado TJBot, nodo shine. Mode: Shine, Color: green, Name: Disgustado.
9. Del apartado TJBot, nodo shine. Mode: Shine, Color: magenta, Name: Miedo.
10. Del apartado TJBot, nodo shine. Mode: Shine, Color: yellow, Name: Alegria.
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 la red con este hashtag se detectará casi en tiempo real en nuestro flujo, y la información contenida en el twit 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 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.

La siguiente tabla describe más precisamente los sentimientos que se pueden encontrar en el análisis de Twitter.

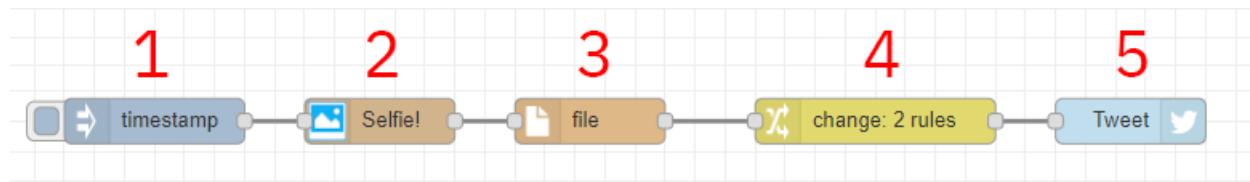
En la tabla siguiente se describen los tonos de finalidad general que puede devolver el servicio. Los tonos con una puntuación inferior a 0.5 se omiten, lo que indica que es poco probable que la emoción se perciba en el contenido. Una puntuación superior a 0.75 indica una alta probabilidad de que el tono se perciba.

Tono / ID	Descripción
Ira <code>anger</code>	Se evoca la ira con una injusticia, un conflicto, una humillación, una negligencia o una traición. Si la ira está activa, el individuo ataca al objetivo, de forma verbal o física. Si ira es pasiva, la persona se enoja de forma silenciosa y siente tensión y hostilidad. (Un tono emocional.)
Miedo <code>fear</code>	El miedo es una respuesta a un peligro inminente. Es un mecanismo de supervivencia que se activa como reacción a un estímulo negativo. El miedo puede ser una advertencia leve o una fobia extrema. (Un tono emocional.)
Alegria <code>joy</code>	La alegría (o felicidad) tiene matices de disfrute, satisfacción y placer. La alegría aporta una sensación de bienestar, paz interior, amor, seguridad y satisfacción. (Un tono emocional.)
Tristeza <code>sadness</code>	La tristeza indica un sentimiento de pérdida y desventaja. Cuando una persona está callada o menos energética o se muestra introvertida, se puede inferir que siente tristeza. (Un tono emocional.)
Analítico <code>analytical</code>	Un tono analítico indica el razonamiento y la actitud analítica de una persona frente a temas. Una persona analítica podría percibirse como intelectual, racional, sistemática, insensible o impersonal. (Un tono del lenguaje.)
Seguro de uno mismo <code>confident</code>	Un tono de autoconfianza indica el grado de seguridad de una persona. Una persona segura podría percibirse como segura, serena, optimista o egoísta. (Un tono del lenguaje.)
Indecisión <code>tentative</code>	Un tono indeciso indica el grado de inhibición de la persona. Una persona indecisa podría percibirse como cuestionable, dubitativa o discutible. (Un tono del lenguaje.)

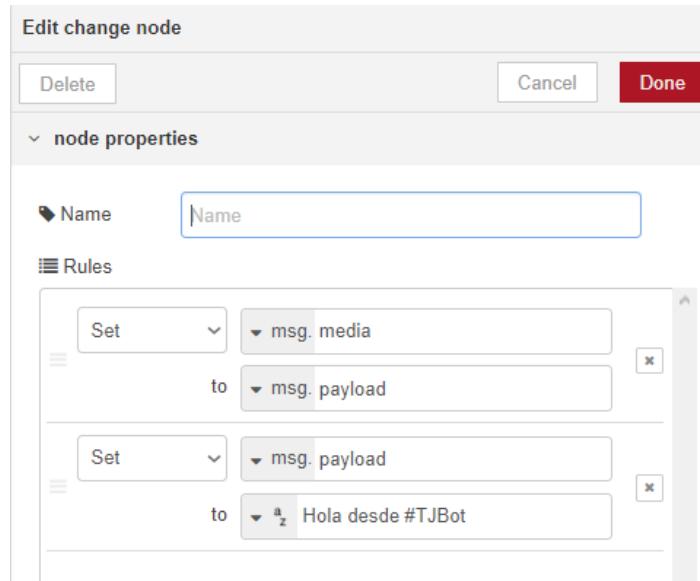
Instrucciones para que TJBot twittee una selfie

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:



1. Del apartado Input nodo inject.
2. Del apartado TJBot, nodo See. **Bot:** maría, **Mode:** Take photo. **Flip:** Horizontal. **Dimensions:** 960x720, **Name:** Selfie.
3. Del apartado storage, el primer nodo file. **Output:** a single buffer object.
4. Del apartado function, nodo change. Configuración:



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.

Anexos y Troubleshooting

Los siguientes anexos contienen información útil para la resolución de problemas variados, y un enfoque más experto de los componentes y tipos de conexión que posee TJBot. En los siguientes subtemas podrán encontrar fallos comunes detectados con sus posibles soluciones, y métodos de conexión para obtener más información del sistema operativo y del servidor Node-RED.

Conexión a TJBot por VNC (Escritorio remoto)

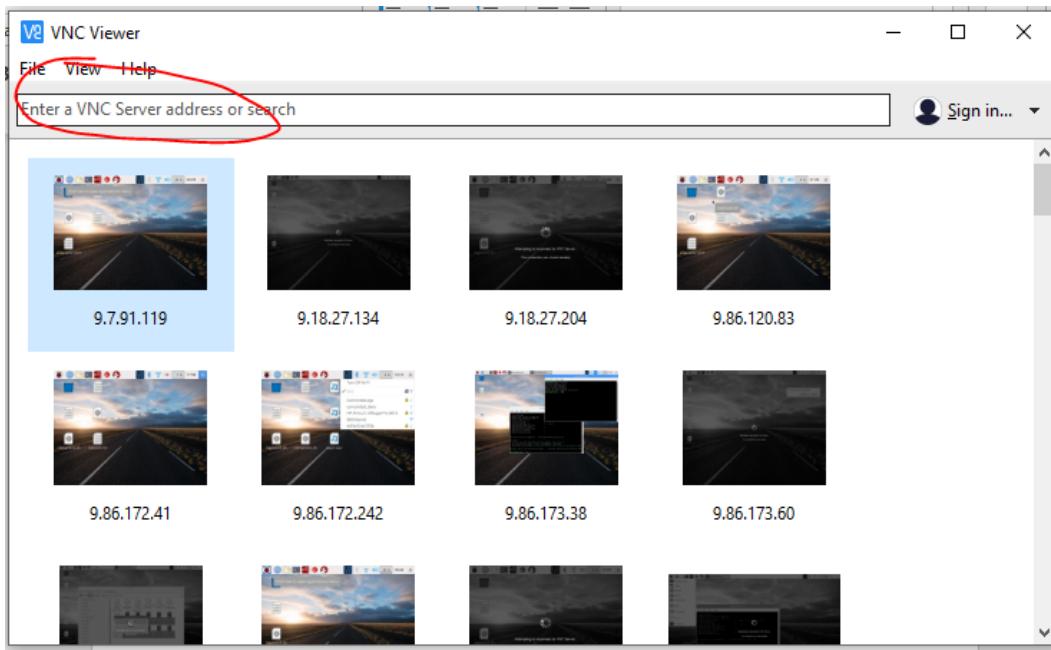
Computación Virtual en Red (Por sus siglas del inglés Virtual Network Computing) es un programa que permite ver e incluso controlar las acciones de una computadora de forma remota, de forma que es necesario tener un cliente y un servidor. El cliente es quien va a observar remotamente otra computadora, mientras que el servidor es la computadora a la que se le puede observar y/o controlar.

Raspbian (y por lo tanto TJBot) tienen por defecto un servidor de VNC al cual es posible conectarse haciendo uso de un programa cliente VNC, la dirección IP de la Raspberry y las credenciales de acceso.

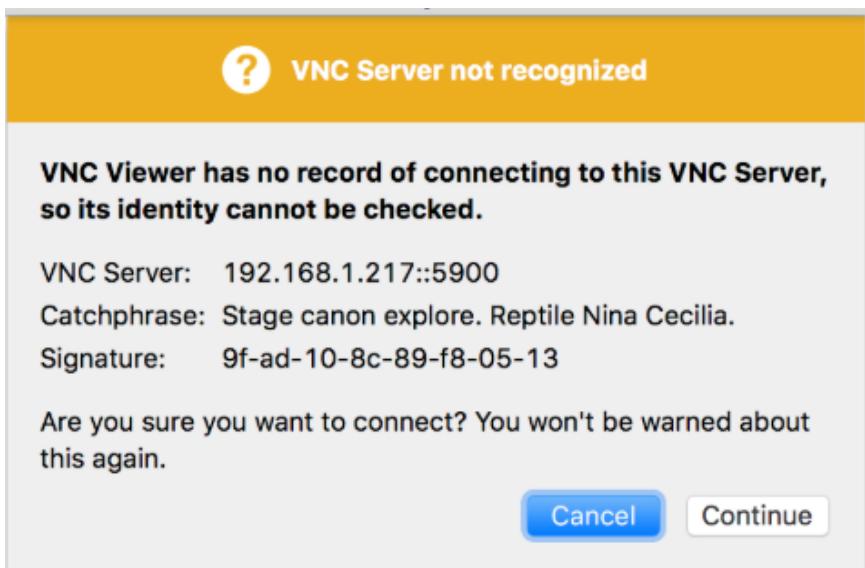
En TJBot hacemos uso de la conexión VNC cuando deseamos explorar la interfaz gráfica, el explorador de archivos o la ventana de comandos y no contamos con una pantalla extra físicamente para conectar a TJBot.

Para lograr una conexión por VNC:

1. Instale el cliente VNC viewer en su computador:
<https://www.realvnc.com/es/connect/download/viewer/>
2. Ejecute el programa, y en el espacio que dice “Enter a VNC Server address...” introduzca la dirección IP del robot.

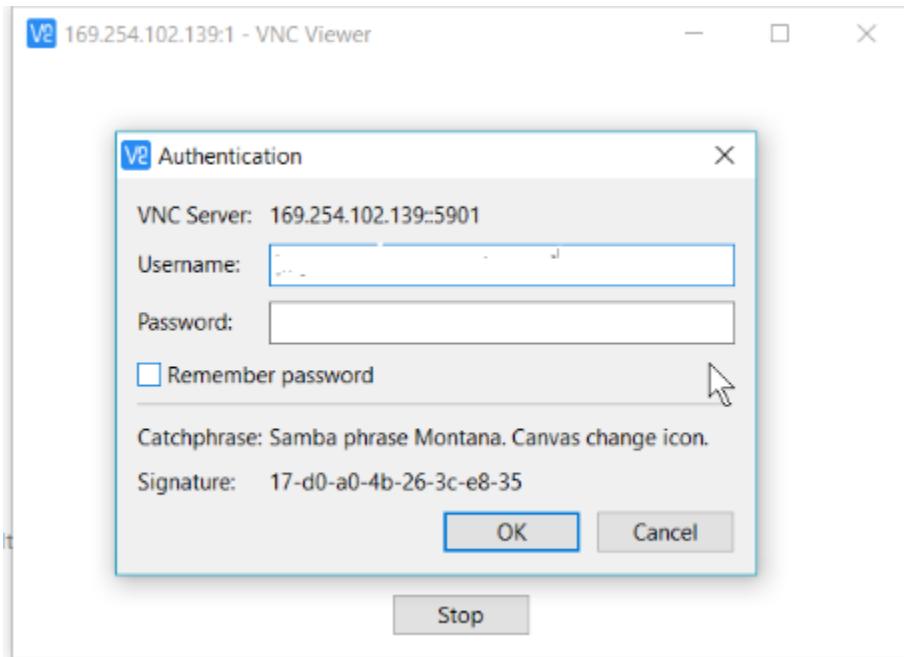


3. Al introducir la dirección aparecerá una ventana de advertencia de que el servidor no es conocido, dar click en continuar.



4. Aparecerá una ventana para iniciar sesión.
 - a. En el caso de TJBot el Username es **pi** y el password es **ibmtjbot**.

- b. En el caso de la Raspberry PI si su contraseña no ha sido cambiada, el usuario es **pi** y el password es **raspberry**.



Tras seguir estos pasos deberás estar dentro de la interfaz gráfica de forma remota, y hacer uso de los componentes con normalidad como si estuvieras conectado físicamente a una pantalla.

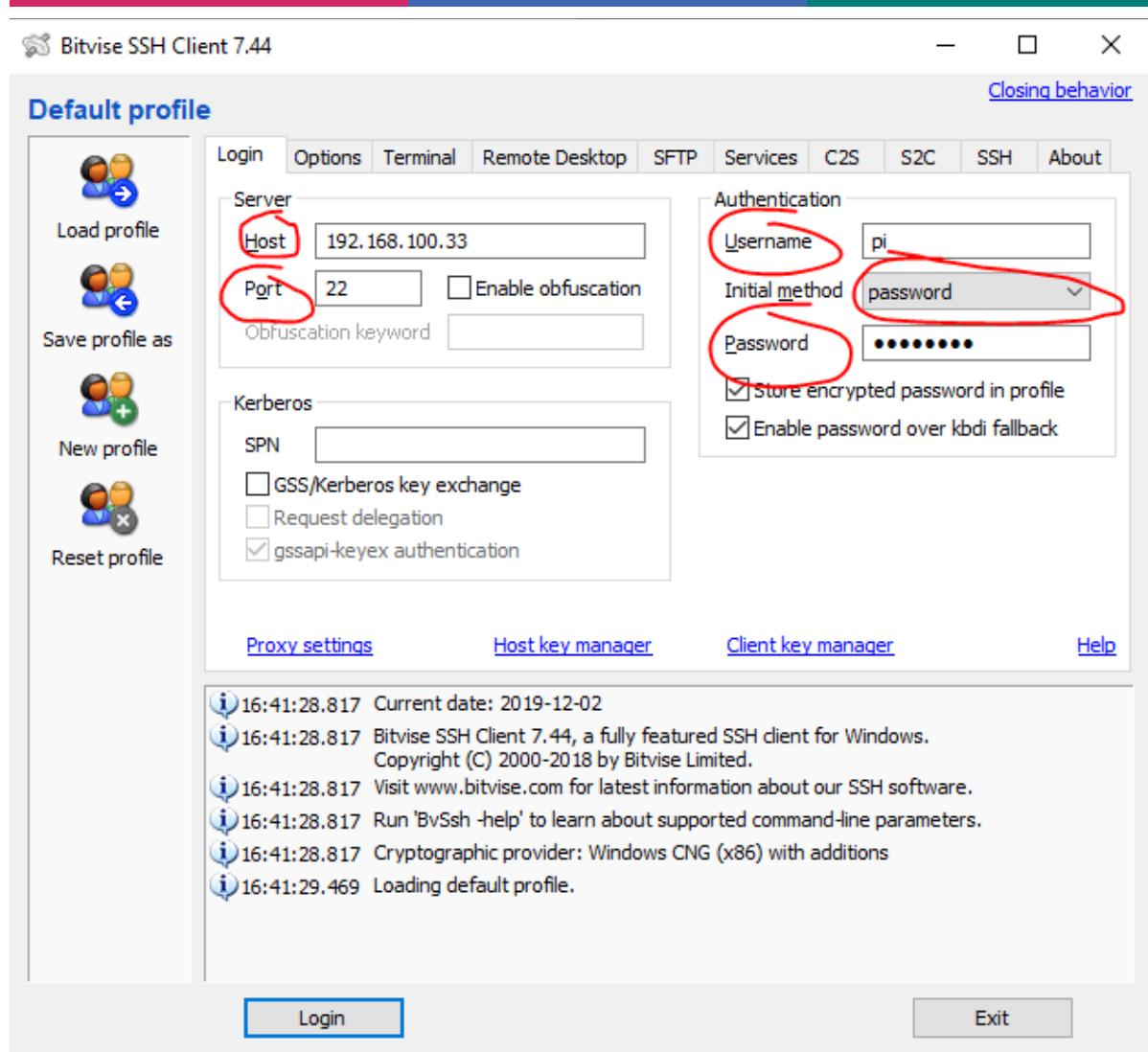
Conexión a TJBot por SSH (Ventana de comandos remota)

Secure Shell (SSH) es un protocolo de comunicación entre computadoras que, de forma similar a VNC, sirve para interactuar de forma remota con una computadora. A diferencia de VNC, la conexión SSH abre únicamente una ventanilla de comandos en lugar de ejecutar la interfaz gráfica.

Abrir solamente la ventana de comandos es una ventaja porque consume menos ancho de banda de internet y nos permite hacer la mayoría de tareas necesarias para la administración de nuestra Raspberry PI.

Para realizar una conexión remota por SSH:

1. Instale algún programa para conexiones por ssh, por ejemplo para Windows Bitvise SSH Server: <https://www.bitvise.com/ssh-client-download>
2. Ejecute el programa e introduzca la siguiente información:



- Host: La dirección IP del robot.
- Port: El puerto de funcionamiento de SSH, por default es el 22.
- Username: **pi**.
- Initial method: **password**.
- Password: En el caso de TJBot **ibmtjbot**, en el caso que no se haya configurado la contraseña default es **raspberry**.
- Dar click en login.

Si la conexión fue exitosa se abrirá automáticamente una ventana de comandos, es la misma ventana que si accediéramos a través de la interfaz gráfica. Esta ventana de comandos será de utilidad para la resolución de algunos problemas comunes en TJBot que exploraremos en los siguientes capítulos.

La ventana de comandos como fuente de resolución de problemas de Node-RED

Cuando accedemos al programa Node-RED lo hacemos a través del navegador web, pero esta no es una página web que se encuentra en internet, es una página web generada por el servidor alojado en el sistema operativo.

Este servidor por default se activa automáticamente cada que se enciende la tarjeta, motivo por el cual podemos acceder a la interfaz sin activar nosotros manualmente el servidor.

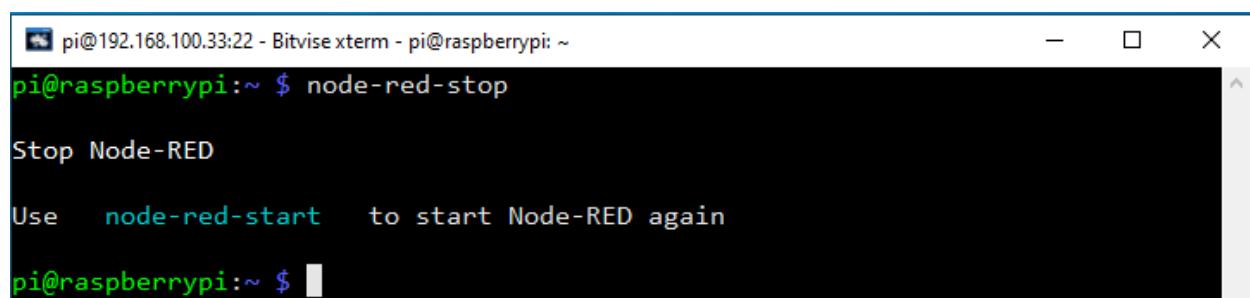
Este servidor de puede iniciar y detener desde la ventana de comandos introduciendo los siguientes comandos:

node-red-start: Este comando iniciará el servidor.

node-red-stop: Este comando detendrá el servidor.

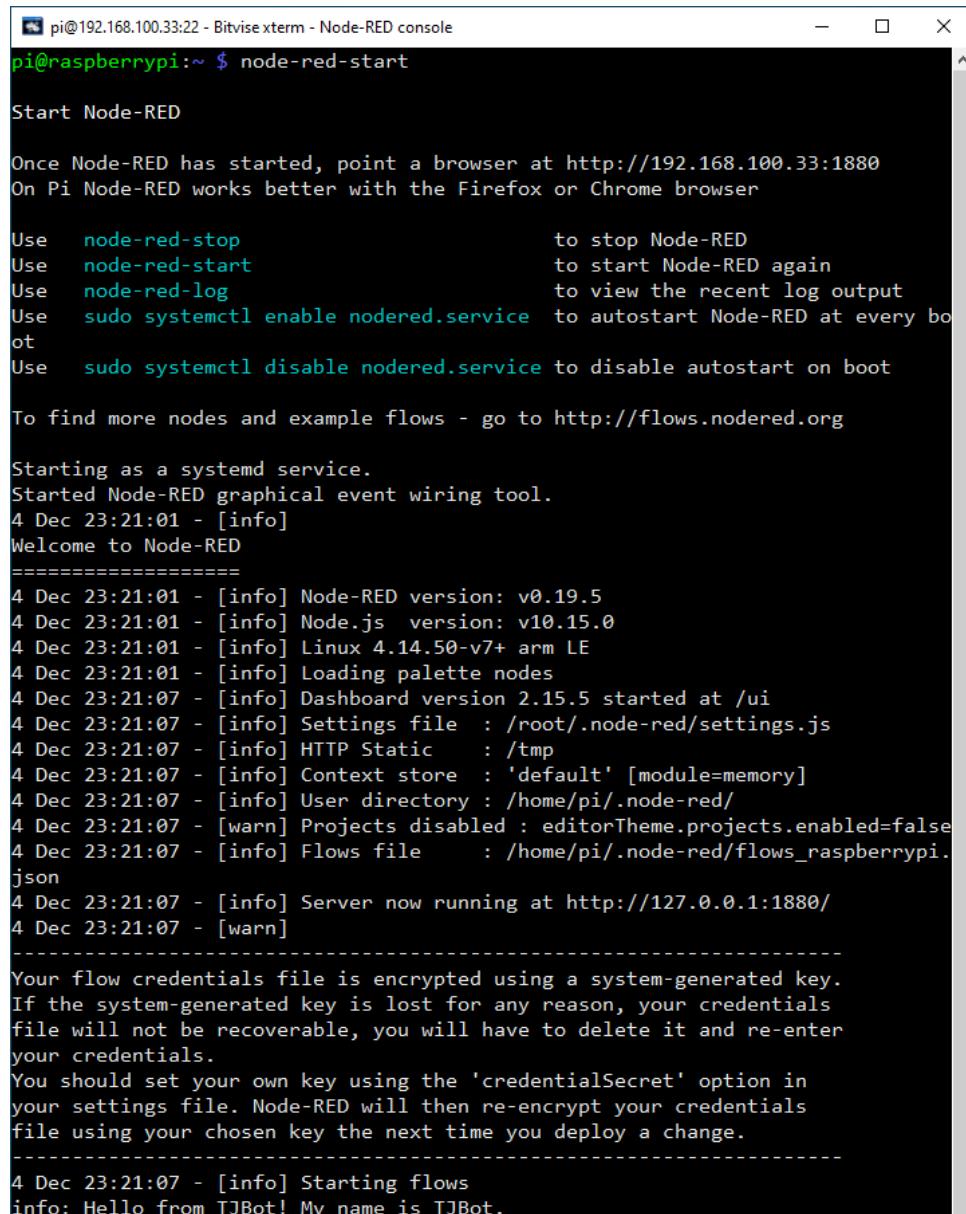
Cuando nos encontramos en esta ventana con el servidor iniciado podemos ver el resultado de la mayoría de las ejecuciones que se realicen en la interfaz Node-RED pero a mayor nivel de detalle, desde ejecuciones fallidas como las válidas y fallos en la configuración, los datos que se arrojan son más útiles que aquellos que se obtienen de la ventana debug de Node-RED. Cuando nos encontramos con un problema en TJBot que no es posible resolver con la interfaz, recurrimos a esta ventana informativa para visualizar el estado de la ejecución del servidor.

Vamos a exemplificar cómo acceder a la ventana. Con TJBot conectado entra a la ventana de comandos (en cualquier método de conexión, es indistinto). Introduce el comando para detener el servidor:



```
pi@192.168.100.33:22 - Bitvise xterm - pi@raspberrypi: ~
pi@raspberrypi:~ $ node-red-stop
Stop Node-RED
Use    node-red-start    to start Node-RED again
pi@raspberrypi:~ $
```

Ahora inicia el servidor, verás información de utilidad. Si todo está bien, los datos que verás son similares a los siguientes:



pi@192.168.100.33:22 - Bitvise xterm - Node-RED console

```
pi@raspberrypi:~ $ node-red-start

Start Node-RED

Once Node-RED has started, point a browser at http://192.168.100.33:1880
On Pi Node-RED works better with the Firefox or Chrome browser

Use  node-red-stop          to stop Node-RED
Use  node-red-start         to start Node-RED again
Use  node-red-log           to view the recent log output
Use  sudo systemctl enable nodered.service to autostart Node-RED at every bo
ot
Use  sudo systemctl disable nodered.service to disable autostart on boot

To find more nodes and example flows - go to http://flows.nodered.org

Starting as a systemd service.
Started Node-RED graphical event wiring tool.
4 Dec 23:21:01 - [info]
Welcome to Node-RED
=====
4 Dec 23:21:01 - [info] Node-RED version: v0.19.5
4 Dec 23:21:01 - [info] Node.js  version: v10.15.0
4 Dec 23:21:01 - [info] Linux 4.14.50-v7+ arm LE
4 Dec 23:21:01 - [info] Loading palette nodes
4 Dec 23:21:07 - [info] Dashboard version 2.15.5 started at /ui
4 Dec 23:21:07 - [info] Settings file  : /root/.node-red/settings.js
4 Dec 23:21:07 - [info] HTTP Static   : /tmp
4 Dec 23:21:07 - [info] Context store  : 'default' [module=memory]
4 Dec 23:21:07 - [info] User directory : /home/pi/.node-red/
4 Dec 23:21:07 - [warn] Projects disabled : editorTheme.projects.enabled=false
4 Dec 23:21:07 - [info] Flows file    : /home/pi/.node-red/flows_raspberrypi.
json
4 Dec 23:21:07 - [info] Server now running at http://127.0.0.1:1880/
4 Dec 23:21:07 - [warn]
-----
Your flow credentials file is encrypted using a system-generated key.
If the system-generated key is lost for any reason, your credentials
file will not be recoverable, you will have to delete it and re-enter
your credentials.
You should set your own key using the 'credentialSecret' option in
your settings file. Node-RED will then re-encrypt your credentials
file using your chosen key the next time you deploy a change.
-----
4 Dec 23:21:07 - [info] Starting flows
info: Hello from TJBot! My name is TJBot.
```

Hay información relevante cuando el servidor inicia.

Comandos para interactuar con el servidor Node-RED.

```
Use  node-red-stop          to stop Node-RED
Use  node-red-start         to start Node-RED again
Use  node-red-log           to view the recent log output
Use  sudo systemctl enable nodered.service to autostart Node-RED at every bo
ot
Use  sudo systemctl disable nodered.service to disable autostart on boot
```

El servicio inició internamente, inició el entorno gráfico y da la bienvenida.

```
Starting as a systemd service.  
Started Node-RED graphical event wiring tool.  
4 Dec 23:21:01 - [info]  
Welcome to Node-RED  
=====
```

Versiones de las tecnologías involucradas, rutas importantes donde se alojan archivos como los flujos, archivos estáticos http, archivo de flujos y el estatus del servidor corriendo.

```
-----  
4 Dec 23:21:01 - [info] Node-RED version: v0.19.5  
4 Dec 23:21:01 - [info] Node.js  version: v10.15.0  
4 Dec 23:21:01 - [info] Linux 4.14.50-v7+ arm LE  
4 Dec 23:21:01 - [info] Loading palette nodes  
4 Dec 23:21:07 - [info] Dashboard version 2.15.5 started at /ui  
4 Dec 23:21:07 - [info] Settings file  : /root/.node-red/settings.js  
4 Dec 23:21:07 - [info] HTTP Static   : /tmp  
4 Dec 23:21:07 - [info] Context store  : 'default' [module=memory]  
4 Dec 23:21:07 - [info] User directory : /home/pi/.node-red/  
4 Dec 23:21:07 - [warn] Projects disabled : editorTheme.projects.enabled=  
4 Dec 23:21:07 - [info] Flows file    : /home/pi/.node-red/flows_raspberryPi.json  
4 Dec 23:21:07 - [info] Server now running at http://127.0.0.1:1880/  
4 Dec 23:21:07 - [warn]  
-----
```

Iniciando flujos y nombre del Bot configurado.

```
4 Dec 23:21:07 - [info] Starting flows  
info: Hello from TJBot! My name is TJBot.  
-----
```

El comando se ejecutará continuamente hasta que cierres la ventana o con el teclado teclees la combinación **ctrl+x**. A partir de este momento las actividades que ejecutes en la interfaz arrojaran información de los procesos a esta ventana. Los siguientes temas abordan problemáticas comunes donde esta ventana de comandos es de utilidad para la resolución de problemas.

Por ejemplo, si iniciamos la actividad de habla de TJBot se despliegan los siguientes datos:

```
===== { filename: '/tmp/tjbot20191114-1274-83349v.oahjv',  
gain: 100,  
debug: true,  
player: 'aplay',  
device: 'plughw:0,0' }
```

Donde podemos ver el archivo de sonido que fue reproducido, la ganancia, si el debug está activado, el comando para reproducir el sonido y el dispositivo de audio utilizado.

TJBot no habla

Si ya corroboraste que las llaves de API y la programación es correcta, sigue estos pasos y pistas para averiguar cuál es el problema.

Antes que nada entra a la ventana de comandos e inicia el servidor para ver los mensajes de error que podrían aparecer.

1. *Cuando inicio la actividad de habla solo se escucha un “pop” breve en la bocina.*
Esto ocurre porque la credencial introducida en TJBot está mal escrita, corrobora nuevamente tu credencial.

2. *No hay señales raras.*

Si no hay pistas del error detén el servidor y vuelve a iniciar, presta atención a los mensajes de la ventana de comandos.

El siguiente mensaje aparece cuando las credenciales son erróneas:

```
4 Dec 23:57:10 - [info] Starting flows
info: Hello from TJBot! My name is TJBot.
4 Dec 23:57:11 - [info] Started flows
error: unable to retrieve TTS voices message={"context":{"requestId":"c55477dc9fc54c01b5af5c14e8b66387","requestType":"incoming.Identity_Token","userAgent":"NotSet","url":"https://iam.bluemix.net","instanceId":"iamid-4.6-6958-c1756bd-7988bdc887-wt8c4","threadId":"3a8131","host":"iamid-4.6-6958-c1756bd-7988bdc887-wt8c4","startTime":"04.12.2019 23:57:12:039 GMT","endTime":"04.12.2019 23:57:12:094 GMT","elapsedTime":55,"locale":"en_US","clusterName":"iam-id-prdal12-8brn"},"errorCode":"BXNIM0415E","errorMessage":"Provided API key could not be found"}, stack=Error: {"context":{"requestId":"c55477dc9fc54c01b5af5c14e8b66387","requestType":"incoming.Identity_Token","userAgent":"NotSet","url":"https://iam.bluemix.net","instanceId":"iamid-4.6-6958-c1756bd-7988bdc887-wt8c4","threadId":"3a8131","host":"iamid-4.6-6958-c1756bd-7988bdc887-wt8c4","startTime":"04.12.2019 23:57:12:039 GMT","endTime":"04.12.2019 23:57:12:094 GMT","elapsedTime":55,"locale":"en_US","clusterName":"iam-id-prdal12-8brn"},"errorCode":"BXNIM0415E","errorMessage":"Provided API key could not be found"}
    at Request._callback (/home/pi/.node-red/nodes/node-red-contrib-tjbot/node_modules/ibm-cloud-sdk-core/lib/requestwrapper.js:113:21)
    at Request.self.callback (/home/pi/.node-red/nodes/node-red-contrib-tjbot/node_modules/request/request.js:185:22)
    at Request.emit (events.js:182:13)
```

Si bien es mucha la información y mucha de ella es difícil de comprender el error se despliega en formato JSON de llave valor. Si justo después de ver el mensaje donde el Bot se presenta aparece un mensaje como este es que algo está mal configurado. En el ejemplo anterior podemos destacar del primer circulo rojo el texto “error: unable to retrieve TTS voices”, esto indica que no se pudieron obtener las voces del servicio de Watson, a su vez el segundo y tercero circulo indican que el error se debe a que la llave de API no pudo ser encontrada.

La solución consiste simplemente en volver a tu servicio en la plataforma de la nube y obtener nuevamente tu credencial. Si esto no resuelve tu problema te aconsejamos eliminar tu servicio y volverlo a crear.

3. *No hay mensajes de error en la ventana de comandos.*

Si al iniciar la actividad no hay mensajes de error por ningún lado es posible que la Raspberry haya seleccionado la tarjeta de sonido equivocada.

Raspberry tiene varias tarjetas de sonido, gracias a eso puede transmitirlo a través del cable HDMI, puertos USB, Jack o incluso los pines GPIO; el sistema operativo de TJBot se configuró para que al encender asignara las tarjetas de sonido: la bocina tendrá la tarjeta de sonido 1, y el micrófono la cero. Esta asignación se hace al arranque del sistema operativo, pero en raras ocasiones las tarjetas se invierten asignando al micrófono la tarjeta 1 y a la bocina la tarjeta cero, lo que provoca que el micrófono ni la bocina funcionen.

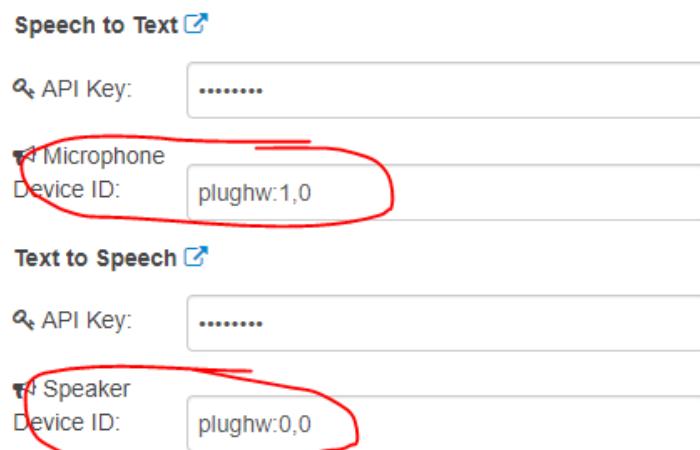
Podemos corroborar la tarjeta de sonido que la raspberry asignó a estos dispositivos por medio de la ventana de comandos, pero para este caso no es necesario usar el servidor, por lo que si estás en el teclea **ctrl+x** para salir.

Para comprobar la tarjeta de sonido asignada a la bocina podemos usar el comando

aplay -l (menos ele), para la tarjeta del microfono el comando es **arecord -l**.

```
pi@raspberrypi:~ $ aplay -l
**** List of PLAYBACK Hardware Devices ****
card 1: Device [USB2.0 Device], device 0: USB Audio [USB Audio]
Subdevices: 1/1
Subdevice #0: subdevice #0
```

La anterior imagen es el resultado del comando para mirar la tarjeta de sonido asignada a la bocina. Hay que mirar donde dice “card” y ver el número, este es el número de tarjeta, “device” más adelante es el tipo de dispositivo. Estos números deben ser configurados en Node-RED en la vista donde configuramos nuestro Bot y agregamos nuestras credenciales.



El valor para modificar son los Microphone Device ID y Speaker Device ID, donde solamente hay que modificar el primer número, que es el que representa la tarjeta de sonido.

Nota:

La mayoría de las veces no es necesario entrar a la ventana de comandos a corroborar la tarjeta asignada ya que las tarjetas de sonido suelen ser uno y cero, por lo que si sospechas que el error pueden ser las tarjetas de sonido ve a la configuración del bot e intercambia los números de tarjeta: El uno por el cero y el cero por el uno, tanto en las credenciales Speech to Text y Text to Speech.

Eliminando programación defectuosa de Node-RED por medio de la ventana de comandos

Es una situación común que el usuario programe una actividad mal planeada, generalmente actividades que se repiten automáticamente, y que estas provoquen que la interfaz comience a trabarse, tarde en cargar o deje de estar disponible por momentos si no es que no vuelve a cargar más.

Por ejemplo: El usuario programa una actividad donde TJBot dice “Hola a todos, cómo están?” e indica que debe repetirse cada segundo. Si el usuario inicia la actividad a los pocos segundos verá un gran problema: El robot está hablando pero la interfaz dejó de responder adecuadamente. La programación recién realizada provocará que TJBot entre en un ciclo de donde no podrá salir fácilmente.

Cuando la actividad incia TJBot envía una petición a la nube enviando el texto, a su vez recibe el resultado de Watson, lo procesa y emite sonido por la bocina por algunos segundos, pero mientras este proceso sucedía otras peticiones cada segundo fueron hechas para el mismo propósito, las ordenes se acumularon y llegó un punto en que el servidor está tan ocupado tratando de atender todas las ordenes que dejó de atender la interfaz gráfica.

Hay muchas actividades similares que podrían llevar al servidor a fallar de esta forma, para detenerlos es necesario eliminar esta programación y la forma fácil de hacerlo es por medio de la interfaz, pero si la interfaz está bloqueada o no responde es necesario:

1. Conectarse a la ventana de comandos por cualquier método de conexión.
2. Detener el servidor: **node-red-stop**
3. El siguiente comando indica que, con privilegios *root*, se elimine el archivo encargado de guardar toda la programación de la interfaz. **Toda programación en la interfaz se perderá**. Para esto introducir el siguiente comando:
sudo rm /home/pi/.node-red/flows_raspberrypi.json
4. Iniciar el servidor: **node-red-start**

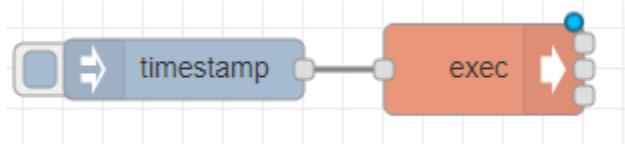
Tras esto puedes acceder nuevamente a la interfaz de TJBot.

Configurar el volumen de TJBot

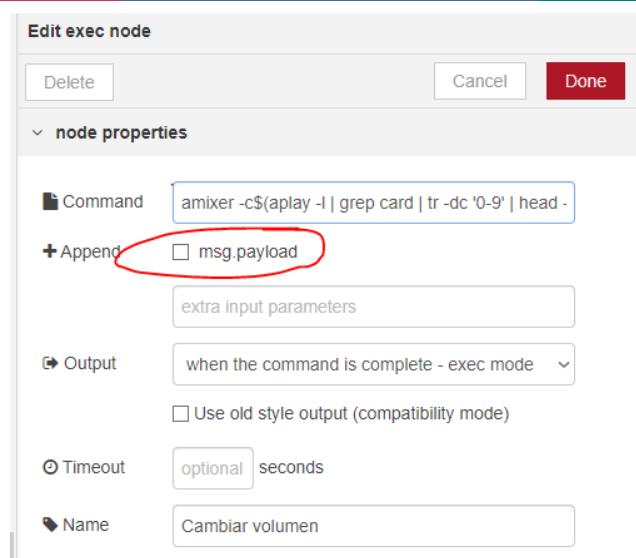
El volumen al que se refiere este anexo es al utilizado cuando TJBot habla o reproduce música. Para configurar el volumen de la dirección IP consulte el anexo “Configurar el volumen de TJBot cuando dicta la dirección IP”.

El volumen puede ser configurado tanto de la interfaz gráfica de TJBot conectado a una pantalla, como por medio de comandos. Cambiarlo por comandos es un proceso más rápido, ya que podemos hacerlo además por la interfaz de Node-RED.

En node-red arrastre los nodos **inject** y **exec**, uniéndolos de la siguiente manera:



El nodo inject no requiere de configuración, mientras que el nodo exec deberá ser configurado de la siguiente manera:



Command:

*amixer -c\$(aplay -l | grep card | tr -dc '0-9' | head -1) set PCM 100%**

Append: Deseleccionar el recuadro.

Name: Cambiar volumen.

*Para el valor de command, reemplazar el número en rojo por el valor en porcentaje del volumen deseado, donde 100% es el máximo y 0% es el mínimo.

El comando que se está introduciendo en command consta de dos partes:

1. El comando es amixer, que nos sirve para seleccionar una tarjeta y configurar su volumen a un valor específico, puede ser en un porcentaje en un rango de 0 a 100% o de forma numérica en un rango del 0 al 255.
2. Dentro del comando amixer se encuentra un subcomando que se ejecutará primero, su procesamiento busca las tarjetas de sonido activas (en este caso encontrará la tarjeta de sonido de la bocina USB) y obtendrá el número de la tarjeta de sonido. Terminado el procesamiento devolverá el número de la tarjeta de sonido activa al comando amixer, para configurar el volumen de esa tarjeta en específico.

Basta con guardar los cambios y ejecutar el flujo para cambiar el volumen del sistema, y modificar el valor del porcentaje para variar el volumen. El comando también puede ser introducido al sistema por medio de la ventana de comandos simplemente copiándolo y pegándolo.

Configurar el volumen de TJBot cuando dicta la dirección IP

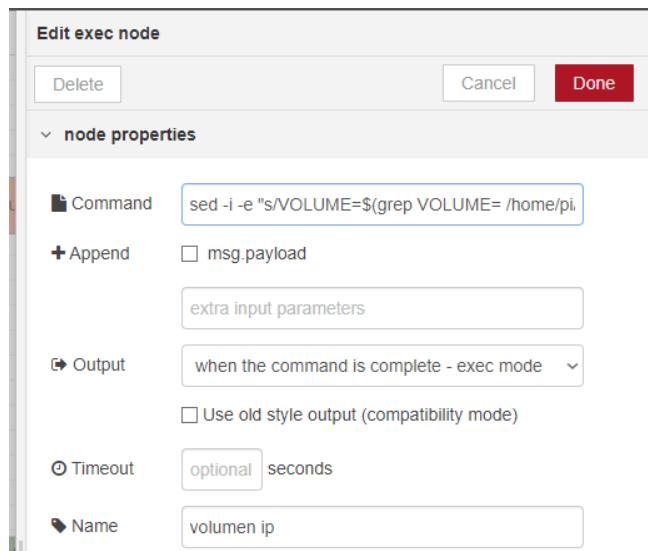
El volumen al que se refiere este anexo es al utilizado cuando TJBot únicamente hace mención de la dirección IP.

El volumen puede ser configurado tanto de la interfaz gráfica de TJBot conectado a una pantalla, como por medio de comandos. Cambiarlo por comandos es un proceso más rápido, ya que podemos hacerlo además por la interfaz de Node-RED.

En node-red arrastre los nodos **inject** y **exec**, uniéndolos de la siguiente manera:



El nodo inject no requiere de configuración, mientras que el nodo exec deberá ser configurado de la siguiente manera:



Command:

`sed -i -e "s/VOLUME=$(grep VOLUME= /home/pi/tjbot_setup/getipaddress.sh | tr -dc '0-9')/VOLUME=20/g" /home/pi/tjbot_setup/getipaddress.sh`

Append: Deseleccionar el recuadro.

Name: Volumen IP.

*Para el valor de command, reemplazar el número en rojo por el valor de volumen en un rango de 0 a 200.

Basta con guardar los cambios y ejecutar el flujo para cambiar el volumen cuando TJBot menciona la dirección IP. El comando también puede ser introducido al sistema por medio de la ventana de comandos simplemente copiándolo y pegándolo.

Eliminar el dictado de la dirección IP

Para que se ejecute el algoritmo dedicado a el dictado de la dirección IP cada que se enciende el sistema se hace uso del recurso de sistema **crontab**, el cual agenda en el reloj del sistema una tarea y que por medio de la notación **@reboot** se indica que se ejecute el comando al arrancar.

Si se desea detener el dictado de la dirección cada que TJBot enciende haga uso de la ventana de comandos:

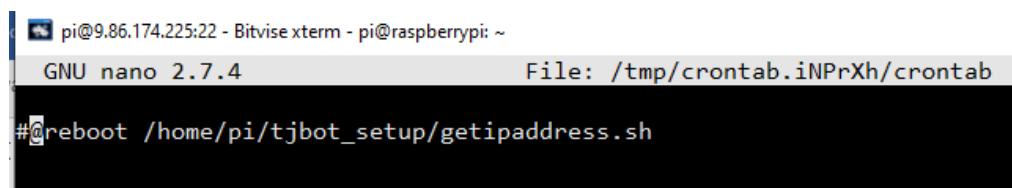
1. Introduzca el comando *crontab -e*. (Si la ventana le pide seleccionar un editor, utilice nano).



pi@9.86.174.225:22 - Bitvise xterm - pi@raspberrypi: ~
GNU nano 2.7.4 File: /tmp/crontab.CqI8Gj/crontab
@reboot /home/pi/tjbot_setup/getipaddress.sh

Crontab

2. Al inicio de la línea, escriba el símbolo de número (#) para deshabilitar la instrucción.



pi@9.86.174.225:22 - Bitvise xterm - pi@raspberrypi: ~
GNU nano 2.7.4 File: /tmp/crontab.iNPrXh/crontab
#@reboot /home/pi/tjbot_setup/getipaddress.sh

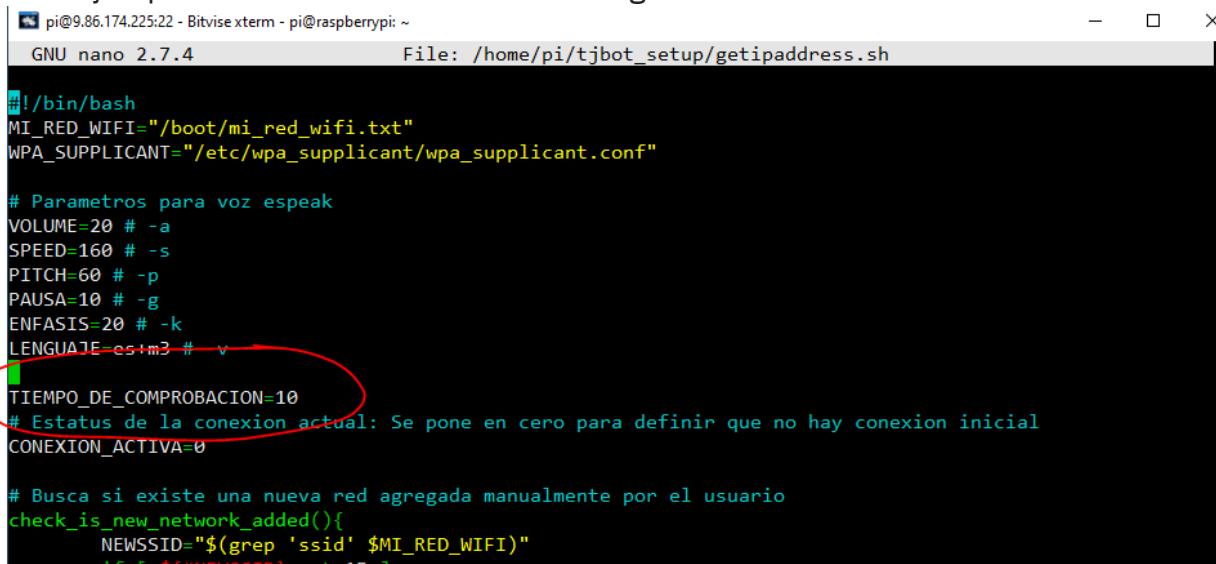
3. Presione la combinación de teclas **ctrl+x** para salir, la tecla **Y** para guardar y **enter** para confirmar.

Tras este proceso habrá deshabilitado la instrucción de repetir la dirección. Si quiere revertir el proceso elimine el símbolo de gato de la línea y vuelva a guardar los cambios.

Modificar el intervalo de comprobación de estatus de red en TJBot

TJBot comprueba cada 5 segundos si tiene una conexión de red válida, solo hasta que tenga una conexión válida dejará de anunciar su estatus. Si deseas ampliar el tiempo de comprobación:

1. Introduce en la ventana de comandos:
sudo nano /home/pi/tjbot_setup/getipaddress.sh
2. En la línea número 13 identifique el valor “TIEMPO_DE_COMPROBACION” y modifique el valor numérico, este se encuentra medido en segundos. En el ejemplo el valor se encuentra en 10 segundos.



```
pi@9.86.174.225:22 - Bitvise xterm - pi@raspberrypi: ~
GNU nano 2.7.4                               File: /home/pi/tjbot_setup/getipaddress.sh

#!/bin/bash
MI_RED_WIFI="/boot/mi_red_wifi.txt"
WPA_SUPPLICANT="/etc/wpa_supplicant/wpa_supplicant.conf"

# Parametros para voz espeak
VOLUME=20 # -a
SPEED=160 # -s
PITCH=60 # -p
PAUSA=10 # -g
ENFASIS=20 # -k
LENGUAJE=es+m3 # -v

TIEMPO_DE_COMPROBACION=10
# Estatus de la conexión actual: Se pone en cero para definir que no hay conexión inicial
CONEXION_ACTIVA=0

# Busca si existe una nueva red agregada manualmente por el usuario
check_is_new_network_added(){
    NEWSSID=$(grep 'ssid' $MI_RED_WIFI)
    if [ ! -z "$NEWSSID" ]; then
        echo "Nueva red agregada: $NEWSSID"
    fi
}
```

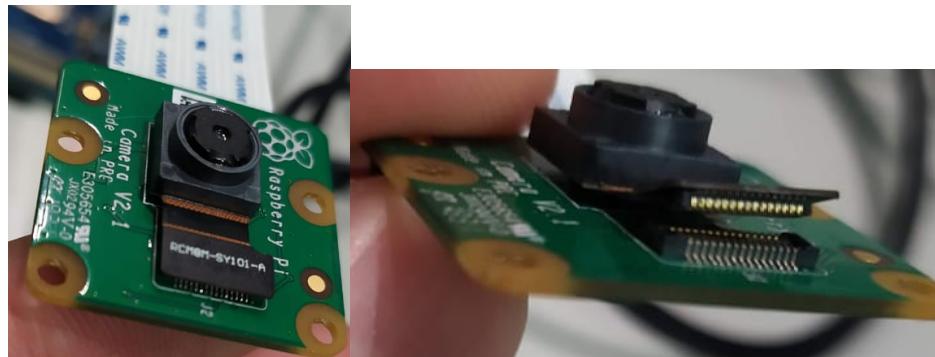
3. Guarde los cambios con la combinación **ctrl+x**, luego **y** para confirmar y **enter** para concluir.

A partir de este punto el tiempo de verificación de red habrá cambiado.

No funciona la cámara de TJBot: Troubleshooting

Si la cámara no está funcionando correctamente, aquí hay un par de cosas que debes comprobar:

1. ¿Está la cinta de la cámara conectada al puerto de cámara y no al de display? Recuerda que la conexión para la cámara en la Raspberry se encuentra justo a un lado del puerto HDMI.
2. La cinta debe estar firmemente conectada a la tarjeta, incluso debes poder sostener la Raspberry PI desde la cámara sin que esta se desconecte.
3. El pequeño conector negro de la cámara debe estar firmemente conectado.



4. Actualiza el sistema por medio de la ventana de comandos con los comandos:
sudo apt update
sudo apt full-upgrade
5. La fuente de alimentación está otorgando suficiente corriente eléctrica? El modulo de cámara añade entre 200 a 250 mA de energía necesaria a la Raspberry PI.

Si todos los pasos anteriores no corrigen el error, entra a la ventana de comandos, detén el servidor e inicialo. Si se muestra algún error identifícalo en alguna de las siguientes situaciones.

Error	Descripción
Error : raspistill/raspivid command not found	Significa que probablemente la actualización falló de alguna manera, vuelve a ejecutar el comando de actualización.
Error : ENOMEM	El módulo de cámara no está iniciando, revisa nuevamente las conexiones.
Error : ENOSPC	<p>La cámara está funcionando sin la suficiente memoria gráfica (GPU).</p> <p>En la ventana de comandos introduce “<i>sudo nano /boot/config.txt</i>”, busca la línea que dice “<i>gpu_mem</i>” y asegúrate de que tenga un valor de al menos 128.</p> <pre># NOOBS Auto-generated Settings: hdmi_force_hotplug=1 start_x=1 gpu_mem=128</pre>

Programación para ir más allá

Aún hay muchos nodos que explorar para sacar el máximo provecho del proyecto, pero estos están fuera del alcance de este manual cuyo propósito es otorgar las herramientas necesarias para seguir explorando con seguridad el mundo de soluciones tecnológicas que el proyecto puede tocar.

Chatbots

Un chatbot es una computadora que sostiene conversaciones con los humanos, a un grado tal que sea difícil distinguir que estamos hablando con una computadora y no con una persona real. Las aplicaciones de estas tecnologías en TJBot permitirán sostener conversaciones con el para diversos propósitos: Una plática, preguntarle información, controlar objetos y realizar reservaciones.

La plataforma de IBM Cloud cuenta con el servicio de Watson Assistant, el cual por medio de una interfaz gráfica te guía paso a paso para la creación de tu propio asistente. La liga a este servicio es la siguiente:

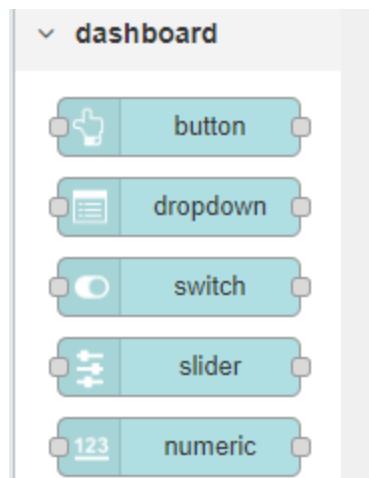
<https://cloud.ibm.com/catalog/services/watson-assistant>

Cuando el asistente es creado, es posible hacer uso de él en plataformas como Messenger, Telegram, incrustarlo en una página web o invocarlo por medio de código haciendo uso de la llave de API.

En la paleta de nodos de Node-RED, en la categoría TJBot podemos encontrar un nodo llamado Conversation, si la credencial y el Workspace fueron correctamente configurados en el Bot este nodo se habilitará de tal manera que al introducir un payload con un texto (podemos introducirlo manualmente o habilitar el escucha del robot) la salida del nodo contendrá la respuesta de nuestro chatbot. Esta respuesta viene etiquetada con las entidades, el contexto y las intensiones del usuario que el chatbot pudo detectar, esto permitirá a TJBot tomar decisiones.

Páginas web

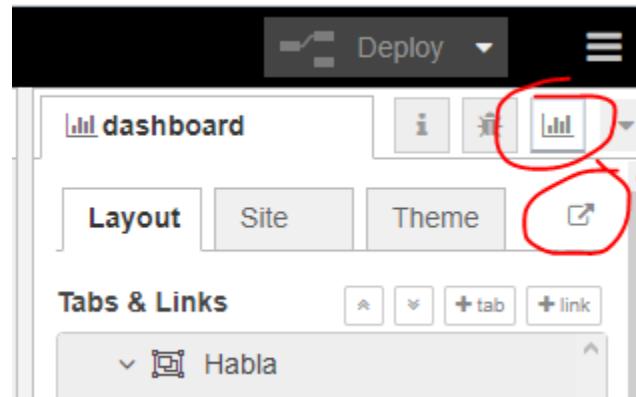
Una de las categorías de nodos más útiles es la de *Dashboard*, nodos enfocados a crear una interfaz de usuario con elementos como botones, formularios, listados, checkboxes o cajas de texto, que en conjunto crean una página web.



Algunos nodos dashboard.

Puede instalar estos nodos buscando la paquetería *node-red-dashboard* en la herramienta para instalar nodos de Node-RED.

Con solo arrastrar los nodos y hacer una configuración inicial de estos, busca en la esquina superior derecha una pequeña pestaña con ícono de gráficos y luego a un botón que te llevará a tu página web.



Acceso a la página web.

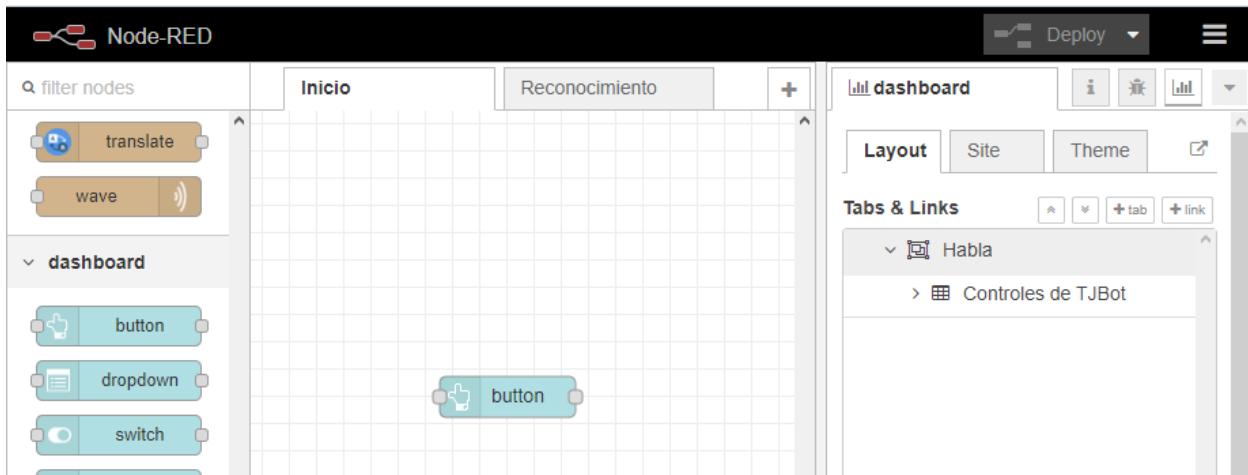
Habla

Controles de TJBot

BUTTON

Interfaz creada con un botón.

En el ejemplo anterior el botón gráfico está ligado a esta programación:



Si a la salida de dicho nodo colocamos otros nodos tales como que TJBot mueva el brazo, estas actividades de iniciarán cuando el usuario de click al botón en la interfaz gráfica.

Una de las utilidades que ofrece la interfaz gráfica son gráficos que toman la información de entrada y la muestran en gráficos de dona o en indicadores de barra.

Cupones para profesores y estudiantes en IBM Cloud

IBM cuenta con el programa de Iniciativa Académica que busca incentivar a estudiantes, investigadores y profesores al uso de las herramientas que ofrece IBM en la nube de potencial utilidad en el campo educativo. Derivado de esto, los estudiantes profesores o investigadores que cuenten con un correo electrónico institucional (otorgado por la escuela) podrán ser acreedores a generar un cupón. Este cupón al ser redimido extiende la capacidad de su cuenta gratuita para poder hacer un uso intensivo de los servicios alojados en la nube, con más capacidad que una cuenta gratuita ordinaria.

Este cupón tiene validez por un año, pero si se sigue contando con este correo institucional es posible renovarlo año tras año para usarlo en proyectos pequeños, tales como TJBot.

Las instrucciones para la obtención de este cupón se encuentran detalladas en la página <https://cognitiveparatodos.mybluemix.net/cupon>

Glosario

API

Significa interfaz de programación de aplicaciones. Una API simplifica la complejidad interna de cualquier tecnología para ser usada de forma sencilla por otros programas de forma independiente al lenguaje de programación.

API Key

También llamado llave de API, es la contraseña necesaria para obtener acceso a un servicio tecnológico que sea ofrecido como API. En IBM Cloud utilizamos las API Keys para obtener acceso a los servicios de la nube de IBM e integrarlo en nuestros proyectos.

Computación Cognitiva

El cómputo cognitivo es la simulación de las capacidades cognitivas de un humano por medio de una computadora, esto le otorga a las computadoras las capacidades cognitivas de escuchar, observar, hablar, razonar o aprender.

Copia de seguridad

Copia íntegra del sistema operativo instalado en la memoria, incluye todos los programas, configuraciones y componentes instalados en el sistema. La copia de seguridad es un archivo que puede guardarse en otras memorias o grabarse para ser replicado.

Deploy

O despliegue, es un botón en Node-RED para guardar los cambios realizados en el espacio de trabajo. Una vez guardados estos cambios la Raspberry puede ser apagada y la programación se conservará.

Ethernet

Es un tipo de conexión alámbrica para obtener internet. La conexión Ethernet se realiza entre un punto de acceso como lo es el módem, un cable ethernet y un puerto o entrada ethernet del dispositivo a conectar a internet.

Hardware

Soporte físico de una computadora, contempla los componentes electrónicos que la constituyen tales como el procesador, el disco duro etc.

IBM Cloud

La nube de IBM es una infraestructura donde se alojan servicios tecnológicos de IBM disponibles para el mundo. Es posible contratar estos servicios para integrar las tecnologías de IBM (Como IBM Watson) a nuestros proyectos, ya que la mayoría de los servicios pueden ser utilizados como una API.

Inteligencia Artificial

Es aquella computadora capaz de razonar y aprender de los datos que se le proveen, permitiéndole tomar decisiones razonadas tal como lo hace el ser humano.

Internet de las Cosas (IoT)

Este concepto se refiere a la interconexión de los objetos a internet, objetos que van desde televisores, focos inteligentes, teléfonos hasta sensores. Estos objetos pueden ser programados para enviar y recibir información de su entorno a internet, de forma que crean una red de comunicación capaz de automatizar tareas. Un ejemplo de la aplicación del IoT son las casas inteligentes.

IP

Es un número que identifica a un dispositivo en una red. Cuando un dispositivo (por ejemplo un celular) se conecta a un punto de acceso (un módem), éste último le asigna una dirección IP para identificarlo y dirigir la información que le corresponde desde esta red hacia internet.

IPv4

Es un protocolo para las direcciones IP que se componen de una serie de 4 octetos separados por un punto, cada octeto se encuentra comprendido entre 0 y 255. Por definición misma del protocolo IPv4, el número total de direcciones IP en el mundo se encuentra limitado a 4.294.967.296. Dirección IPv4 ejemplo: 192.168.255.254

IPv6

El protocolo IPv6 nace para extender la limitación de direcciones de IPv4, pasando de 2^{32} a 2^{128} . Su representación es hexadecimal, cada par de octetos se separa por un punto y su rango va desde 0000 hasta FFFF. Dirección IPv6 ejemplo: 2001:123:4:ab:cde:3403:1:63

JSON

Es un tipo de formato de texto para el intercambio de información entre computadoras, su simplicidad resulta útil para los programadores y eficiente para la programación.

La nube

Hace referencia a la infraestructura tecnológica alojada en internet, generalmente disponible a través de APIs.

LED

LED, también llamado diodo emisor de luz, es un pequeño componente capaz de emitir luz, es usado comúnmente como indicador visual de alguna señal.

Lenguaje de programación

Es una forma de comunicación entre el humano y la computadora. Los lenguajes de programación nacen ante la necesidad de dar instrucciones a la computadora sin necesidad de interactuar con los unos y ceros que componen su software.

NOOBS

New Out Of Box Software, es un programa que sirve para instalar inicialmente en una Raspberry PI un sistema operativo desde cero. Funciona para cuando se instala una Raspberry PI desde cero por primera vez.

Open Source

También llamado de código abierto, son aquellos proyectos que no poseen restricciones para su uso, modificación o integración en otros proyectos.

Proyecto DIY

Del inglés Do It Yourself, hazlo tú mismo, son proyectos sencillos de replicar que no requieren conocimientos especializados.

Raspberry PI

La tarjeta Raspberry PI es una tarjeta de desarrollo de bajo costo, programable, comúnmente utilizada con fines educativos. Su funcionamiento requiere de un sistema operativo y de una fuente de alimentación. El modelo de la tarjeta refleja la generación (hasta el momento hay 4) y la variante.

Raspbian

Es uno de los sistemas operativos disponibles para el funcionamiento de Raspberry PI.

Servomotor

Motor con movilidad de giro en diferentes ángulos que sirve para mover objetos tales como llantas o brazos robóticos.

Sistema operativo

Un sistema operativo es el software principal de una computadora, el cual gestiona los recursos y servicios disponibles para los programas que se ejecutan en el sistema.

Software

Soporte lógico de una computadora, contempla los programas informáticos y su programación.

SSH

Es el nombre de un protocolo para realizar conexiones remotas a la ventana de comandos de un sistema operativo externo. Esto facilita la administración de una computadora como la Raspberry PI sin necesidad de conectarla a una pantalla.

Troubleshooting

Forma sistemática de resolución de problemas.

Ventana de comandos

También llamada terminal o *shell*, es la interfaz más simple entre una persona y una computadora, ésta facilita la administración de recursos y permite configurar a profundidad un sistema operativo. Es importante tener cuidado con los comandos introducidos aquí, pues las configuraciones erróneas pueden llevar a la descompostura de componentes e incluso al del sistema operativo mismo.

VNC

Es un programa que permite controlar la interfaz gráfica de una computadora de forma remota. VNC se compone de dos partes: La computadora servidor que es la que va a ser controlada, y la computadora cliente que va a ser la controladora.

Watson IBM

Watson es una supercomputadora que posee Inteligencia Artificial capaz de responder preguntas en lenguaje natural. Esto conlleva capacidades como el escucha, habla, observar, comprensión del lenguaje humano, razonamiento, entre otros. Watson es accesible a través de la nube de IBM por medio de APIs.

WinRAR

Programa que empaqueta (comprime) y descomprime archivos para que sean compartidos fácilmente.