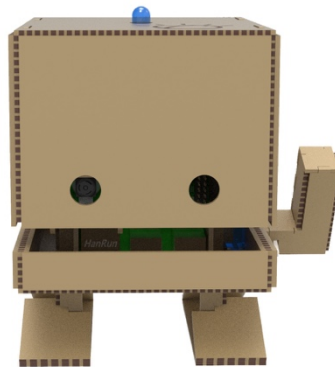


2.PROYECTO: TJBOT

En esta sección vamos a juntar todo lo aprendido anteriormente para construir un **TJBot** con el que poder mantener una conversación. [6]

Para ello vamos a utilizar una **Raspberry Pi3**, la cual conectaremos a los servicios de conversación de Watson y así poder hacer hablar a nuestro robot.



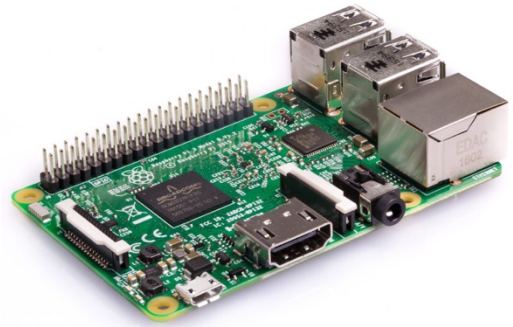
El objetivo final será hacer que nuestro TJBot sea capaz de mantener una conversación con nosotros y obedezca algunas órdenes.

1.1. CÓMO CREAR A TJBOT

1.1.1. RAPSBERRY PI3

Raspberry Pi es un computador de placa única.

Para trabajar con la **Raspberry Pi3** vamos a necesitar un monitor, un ratón, un teclado y una fuente de alimentación para trabajar con ella. También debemos tener una tarjeta micro SD con el sistema operativo instalado, en este caso trabajaremos con el sistema operativo **Raspbian**.



Para crear nuestro TJBot hace falta conectar algunos componentes más a nuestra placa:

- Un micrófono
- Un altavoz
- Una cámara
- Un servomotor
- LED

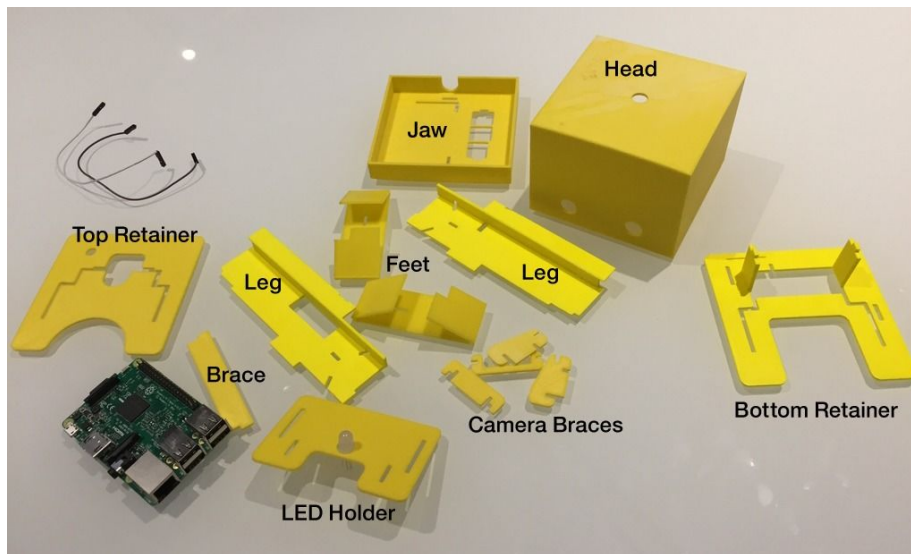
1.2. MONTANDO A TJBOT

Para comenzar a trabajar con **TJBot**, lo primero que debemos hacer es obtener las piezas para poder montarlo.

Este robot es **DY** (Do it yourself) y puede conseguirse imprimiendo las piezas en una impresora **3D**. Las piezas que se deben imprimir pueden encontrarse en la siguiente web:

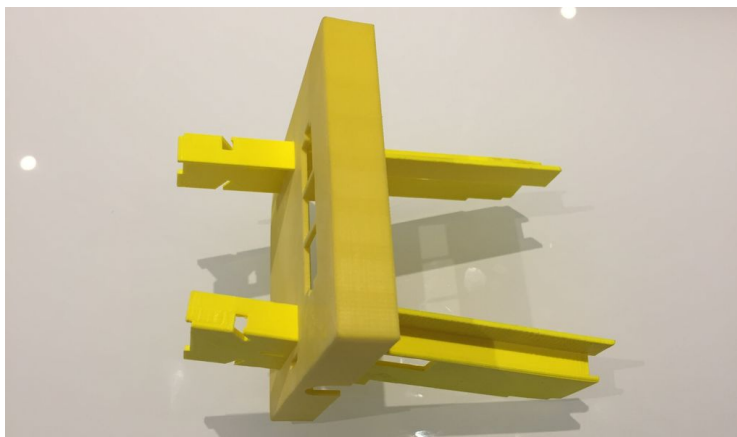
<https://ibmtjbot.github.io/#gettj>

Las piezas que conseguiremos serán las siguientes:

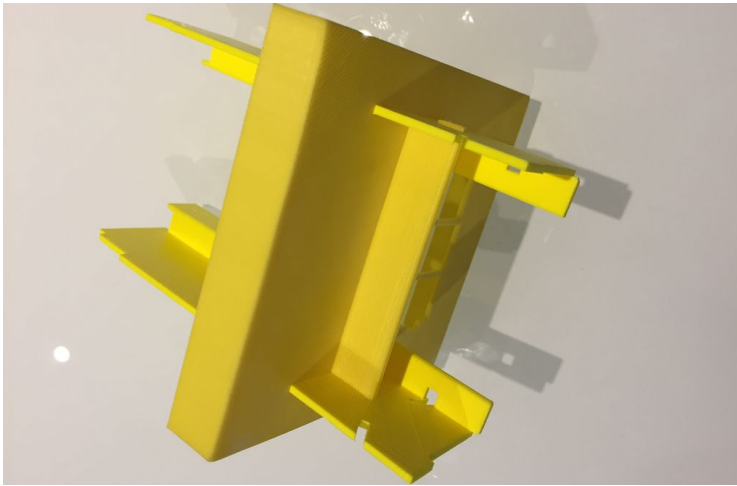


¿Cómo lo montamos?

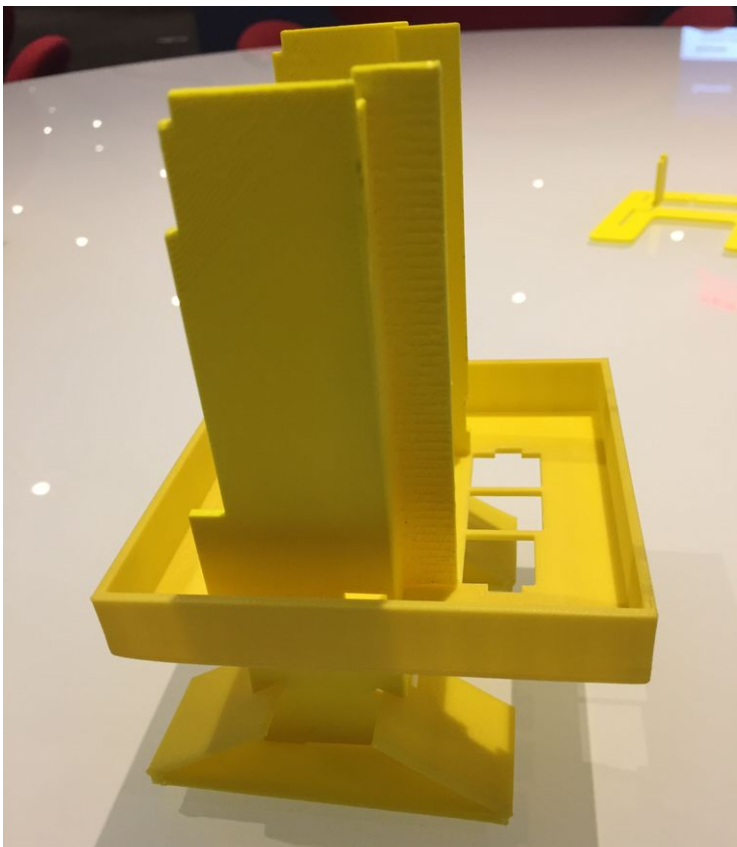
Primero, insertamos las piernas en la mandíbula. Las piernas entran en los agujeros en forma de L en la mandíbula. Inserta desde la parte superior de la mandíbula hacia abajo.



Después, cogemos la pieza “leg brace”. Se encuentra en los dos orificios rectangulares en las patas debajo de la mandíbula. Mantiene las piernas rectas y soporta la mandíbula. Debemos asegurar que esté orientada de modo que haga contacto con la parte inferior de la mandíbula.

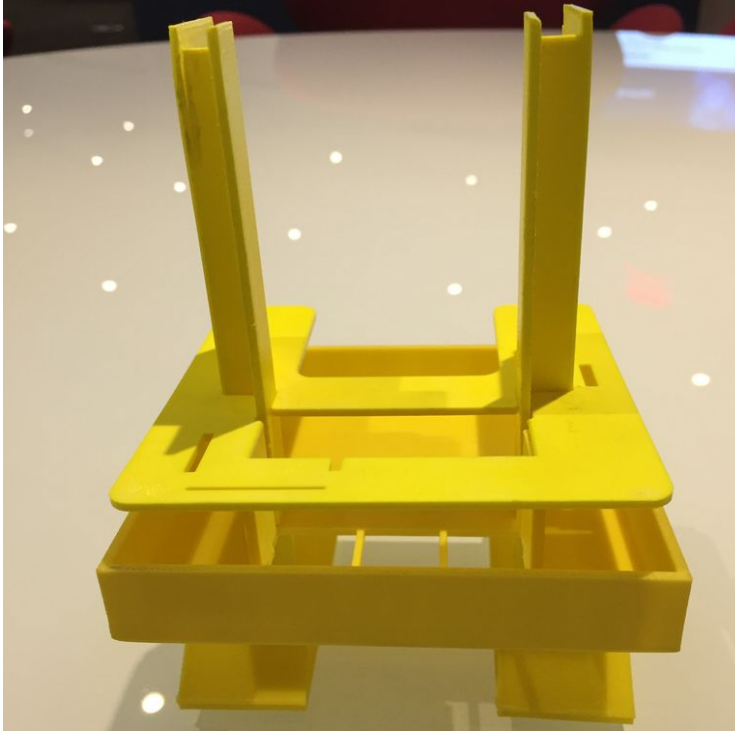


Ahora, cogemos los pies y los insertamos en las muescas de la pierna como se muestra a continuación. Inserta los pies desde el exterior de cada pierna.



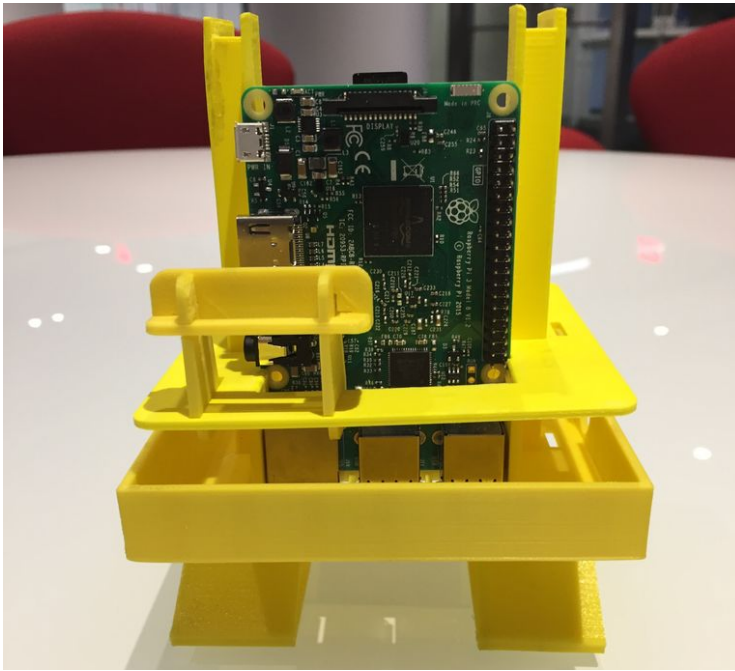
Si tenemos un servo motor, es hora de insertarlo en el orificio del lado izquierdo en la mandíbula.

Cogemos el retenedor inferior y lo deslizamos hacia abajo a través de los agujeros en forma de L en las patas. Los brazos de soporte en la parte inferior de este retenedor deben caber en las muescas de acoplamiento en la mandíbula.

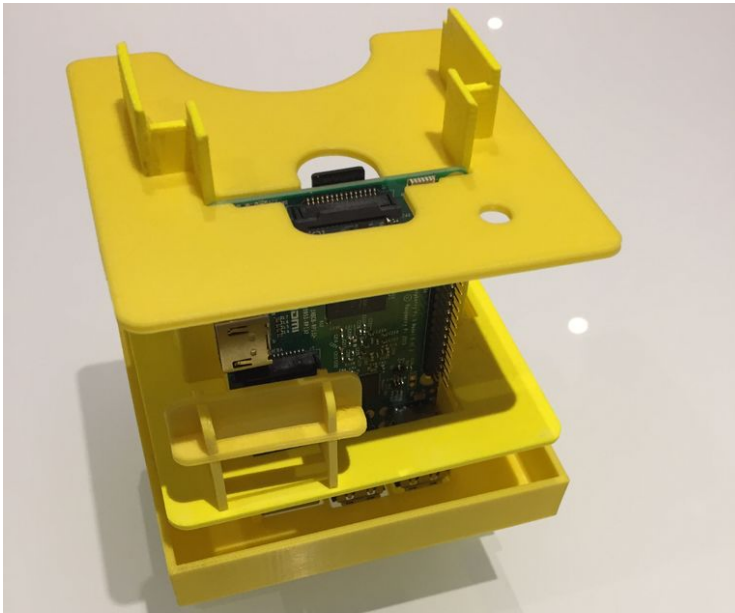


A continuación, insertaremos la Raspberry Pi. Se monta boca abajo, con sus puertos expuestos a través de los tres agujeros en la mandíbula.

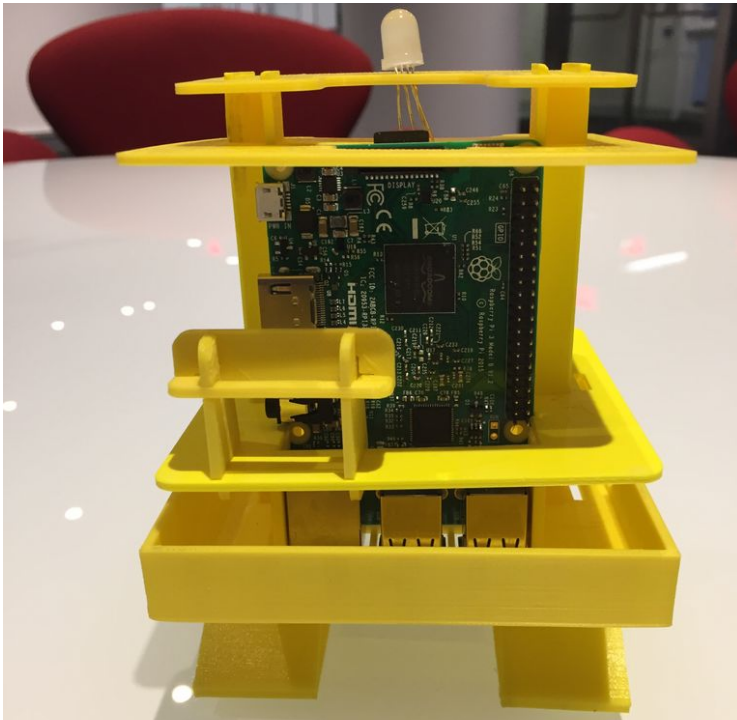
Si tenemos una cámara: tomamos el soporte de montaje de la cámara. Deslizamos los dos montajes laterales de la cámara en las muescas en la parte delantera del retenedor. Deslizamos la cámara Pi en el soporte y luego agregue los reforzadores superior y frontal para mantener la cámara ajustada.



El siguiente retenedor es el que tiene el pequeño círculo grabado en él. Lo añadimos a TJBot deslizándolo a través de los orificios en forma de L, con el círculo pequeño orientado hacia la derecha. Deslizamos el retenedor hacia abajo hasta que se encuentre con la parte superior de la Raspberry Pi.



Insertamos el LED en el orificio central del retenedor pequeño.



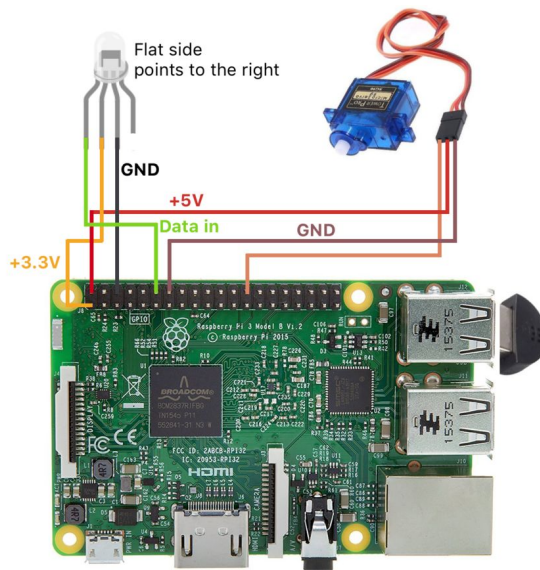
Por último, colocamos la cabeza.



¡Ya está listo nuestro TJBot!

1.2.1. CÓMO CONECTAR LOS COMPONENTES

A continuación, se muestra un esquema de la conexión de los diferentes componentes electrónicos que forman parte de nuestro TJBot.



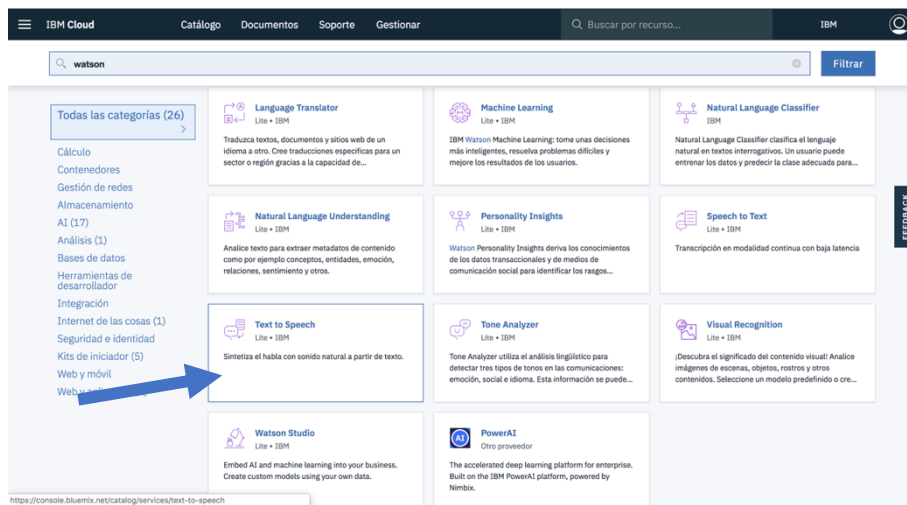
1.3. SERVICIOS DE WATSON

Para realizar el proyecto final del TJBot, vamos a utilizar tres servicios de Watson: Text to Speech, Speech to Text y Watson Assistant.

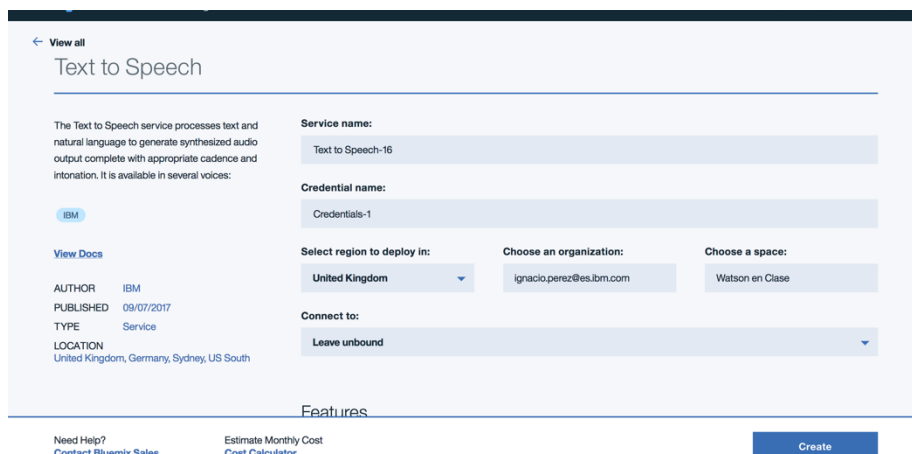
1.3.1. TEXT TO SPEECH

El servicio Text to Speech proporciona una interfaz de programación de aplicaciones que utiliza las capacidades de síntesis de voz de IBM para convertir texto escrito en voz de sonido natural.

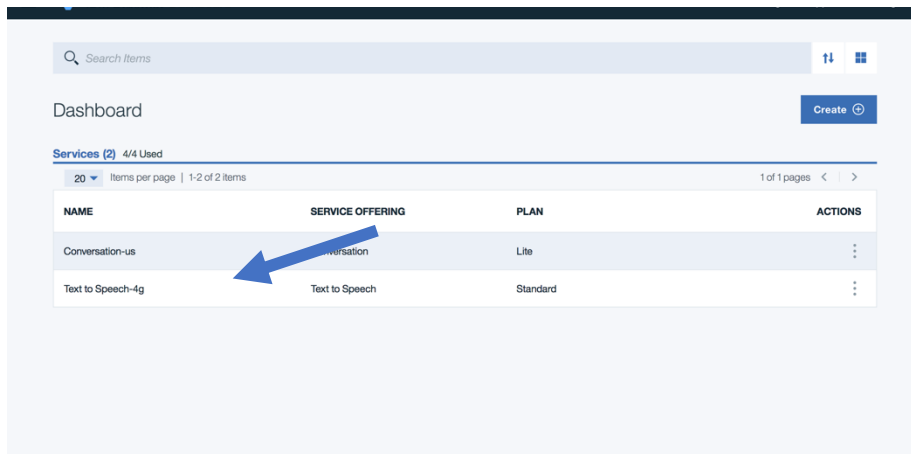
Para crear un servicio Text to Speech de Watson, debemos buscarlo en el catálogo de IBM Cloud.



Una vez seleccionado el servicio deseado, le damos al botón de “Crear” para poder empezar a trabajar.



Cuando ya tenemos nuestro servicio creado, podemos acceder a él desde nuestro Dashboard.



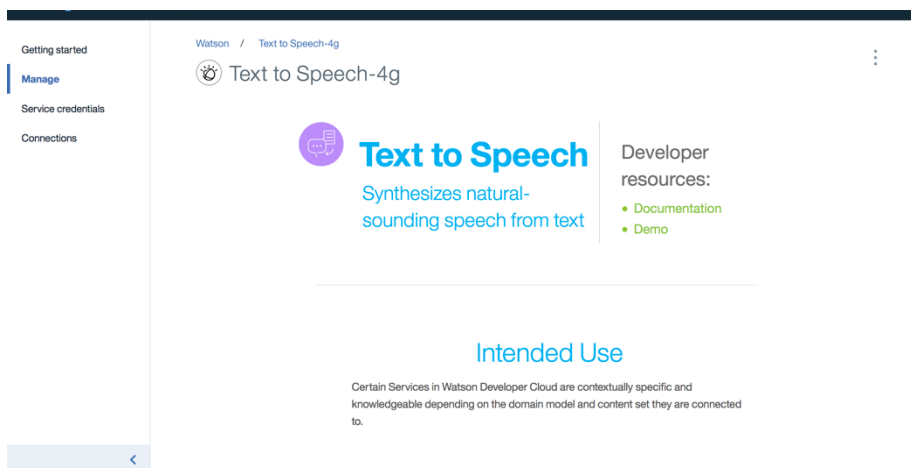
Search Items

Dashboard Create

Services (2) 4/4 Used

20 Items per page | 1-2 of 2 items 1 of 1 pages

NAME	SERVICE OFFERING	PLAN	ACTIONS
Conversation-us	Conversation	Lite	⋮
Text to Speech-4g	Text to Speech	Standard	⋮



Watson / Text to Speech-4g

Text to Speech-4g

Text to Speech
Synthesizes natural-sounding speech from text

Developer resources:

- Documentation
- Demo

Intended Use

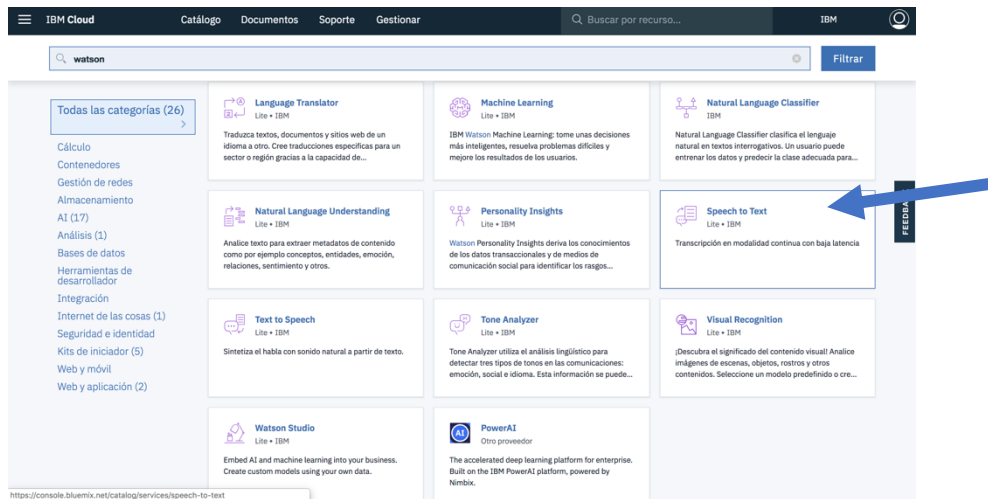
Certain Services in Watson Developer Cloud are contextually specific and knowledgeable depending on the domain model and content set they are connected to.

En el apartado **Service credentials** (credenciales del servicio) podemos ver nuestro usuario y contraseña, que necesitaremos incluir más adelante cuando configuremos el servicio Text to Speech en nuestra Raspberry Pi3.

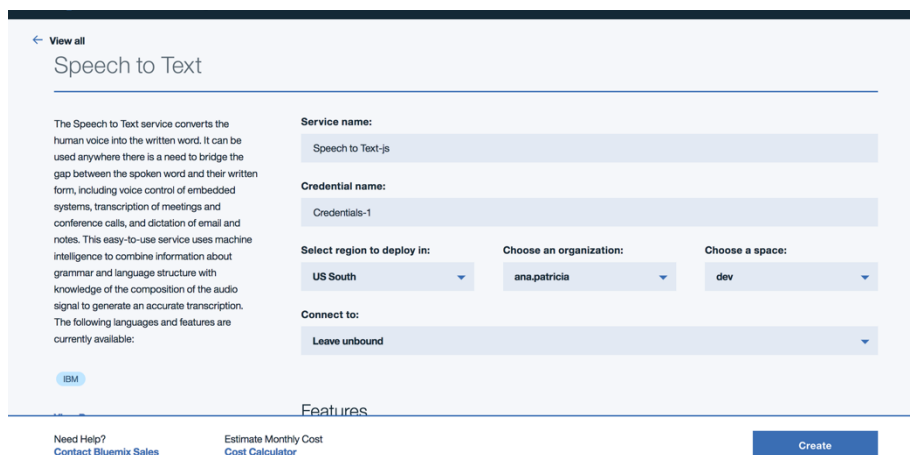
¡Hemos creado nuestro servicio Text to Speech!

1.3.2. SPEECH TO TEXT

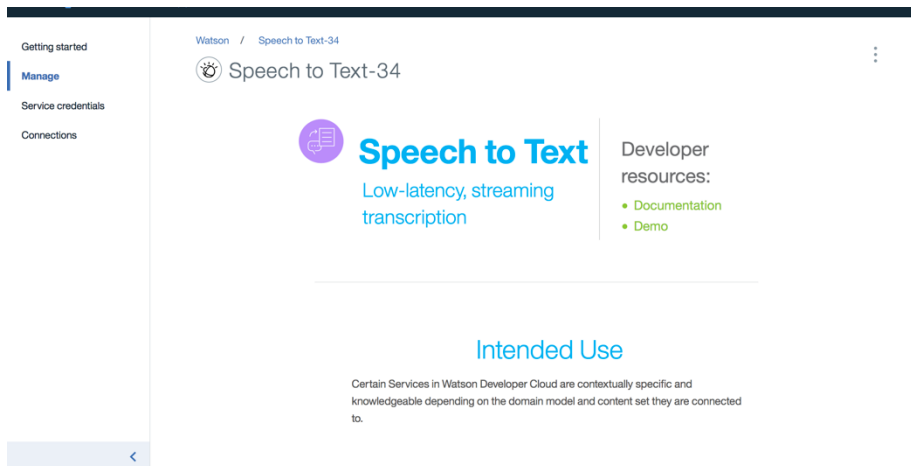
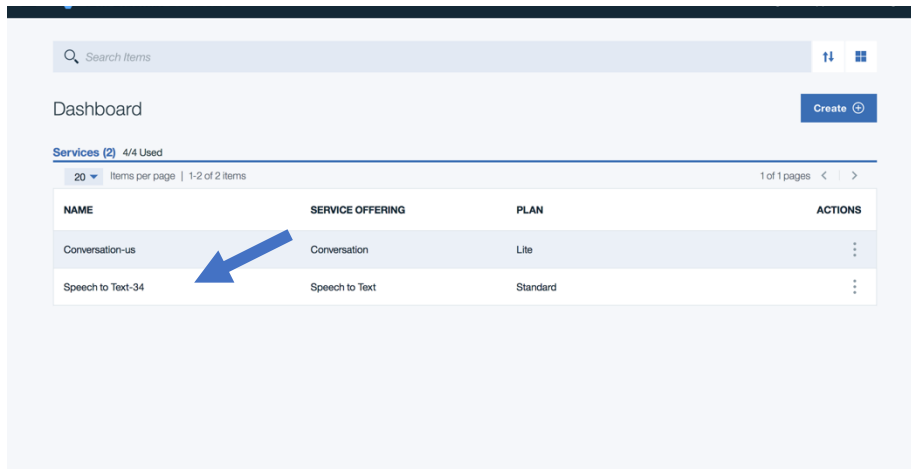
Para crear un servicio Speech to Text de Watson, debemos buscarlo en el catálogo de IBM Cloud.



Una vez seleccionado el servicio deseado, le damos al botón de “Crear” para poder empezar a trabajar.



Cuando ya tenemos nuestro servicio creado, podemos acceder a él desde nuestro Dashboard.



En el apartado **Service credentials** (credenciales del servicio) podemos ver nuestro usuario y contraseña, que necesitaremos incluir más adelante cuando configuremos el servicio Speech to Text en nuestra Raspberry Pi3.

¡Hemos creado nuestro servicio Speech to Text!

1.3.3. ASSISTANT

Para crear un servicio de Watson Assistant, debemos buscar el servicio deseado en el catálogo de IBM Cloud.

IBM Cloud Catálogo Documentos Soporte Gestionar Buscar por recurso... IBM

Catálogo

watson Filtrar

Todas las categorías (26)

- Cálculo
- Contenedores
- Gestión de redes
- Almacenamiento
- AI (17)
- Análisis (1)
- Bases de datos
- Herramientas de desarrollador
- Integración
- Internet de las cosas (1)
- Seguridad e identidad
- Kits de iniciador (5)
- Web y móvil
- Web y aplicación (2)

AI

Watson Assistant (anteriormente Watson Conversation)
Lite • IBM

Añada una interfaz de lenguaje natural a su aplicación para automatizar interacciones con los usuarios finales. Las aplicaciones comunes incluyen agentes...

Discovery
Lite • IBM

Desbloquee el valor oculto de los datos para encontrar respuestas, supervisar tendencias y hacer emerger patrones con el motor de información nativ...

Language Translator
Lite • IBM

Traduzca textos, documentos y sitios web de un idioma a otro. Cree traducciones específicas para un sector o región gracias a la capacidad de...

AI OpenScale
Lite • IBM

IBM AI OpenScale is an enterprise-grade environment for AI infused applications that provides enterprises with visibility into how AI is being built, used, and...

Knowledge Catalog
Lite • IBM

Discover, catalog, and securely share enterprise data.

Machine Learning
Lite • IBM

IBM Watson Machine Learning: tome unas decisiones más inteligentes, resuelva problemas difíciles y mejore los resultados de los usuarios.

Compare Comply
IBM • Beta

Process governing documents to convert, identify, classify, and compare important elements

Knowledge Studio
Lite • IBM

Enseñe a Watson el lenguaje de su dominio.

Natural Language Classifier
IBM

Natural Language Classifier clasifica el lenguaje natural en textos interrogativos. Un usuario puede entrenar los datos y predecir la clase adecuada para...

https://console.bluemix.net/catalog/services/watson-assistant-formerly-conversation

Una vez seleccionado el servicio Assistant, le damos al botón de “Crear” para poder empezar a trabajar.

IBM Cloud Catálogo Documentos Soporte Gestionar Buscar por recurso... IBM

Ver todo

Watson Assistant (anteriormente Watson Conversation)

Lite • IBM

Añada una interfaz de lenguaje natural a su aplicación para automatizar interacciones con los usuarios finales. Las aplicaciones comunes incluyen agentes virtuales y bots de conversación que se pueden integrar y comunicar en cualquier canal o dispositivo. Prepare el servicio Watson Assistant a través de una aplicación web fácil de utilizar, diseñado para que pueda crear rápidamente flujos de conversación natural entre las aplicaciones y los usuarios, y desplegar soluciones escalables y económicas.

[Ver documentos](#) [Condiciones](#)

AUTOR	IBM
PUBLICADO	20/11/2018
TIPO	Servicio

Nombre del servicio:
Watson Assistant (anteriormente Watson Conversa-vj)

Seleccione una región/ubicación de despliegue:
Dallas

Seleccione un grupo de recursos:
Default

Planes de tarifas Los precios mensuales que se muestran son para el país o región: **España**

PLAN	CARACTERÍSTICAS	TARIFAS
✓ Lite	10.000 llamadas de API al mes Hasta 5 espacios de trabajo Hasta 100 intenciones Hasta 25 entidades *POST /solo llamadas a método de mensaje	Gratuito

El plan Lite le inicia con 10.000 llamadas de API al mes sin ningún coste. Y cuando actualiza a un plan de pago, conserva todas las

¿Necesita ayuda?
[Póngase en contacto con el soporte de IBM Cloud](#)

Estimar coste mensual
[Calculadora de costes](#)

Crear

Cuando ya tenemos nuestro servicio creado, podemos acceder a él desde nuestro Dashboard.

Una vez la página principal de nuestro servicio, debemos buscar el botón “Launch tool” que nos llevará a la herramienta en la que configuraremos y desarrollaremos nuestra conversación.

En el apartado **Service credentials** (credenciales del servicio) podemos ver nuestro usuario y contraseña, que necesitaremos incluir más adelante cuando configuremos el servicio Assistant en nuestra Raspberry Pi3.

1.4. CONFIGURAR A TJBOT

1.4.1. CONFIGURAR LA RASPBERRY PI 3

Como ya hemos mencionado, Raspberry Pi es similar a un computador completo, lo que significa que se necesita un monitor, ratón y teclado para utilizarla. [1]

Se puede conectar a un televisor a través de un cable HDMI.

También será necesario conectar la Raspberry a la Wifi.

En la mayoría de los kits Pi, la tarjeta SD ya está precargada con una imagen del sistema operativo Raspberry Pi. Se debe colocar la tarjeta SD en la Pi, encender la Pi y seguir las instrucciones en la pantalla para completar la instalación del sistema operativo.

Una vez tenemos todo preparado, hay que configurar los paquetes para empezar a trabajar.

Abrimos un **terminal** en el Pi y ejecutamos los siguientes comandos para **instalar** la última versión de **Node.js** y **npm** (Node Package Manager). Necesitará estos paquetes más tarde para ejecutar su código.

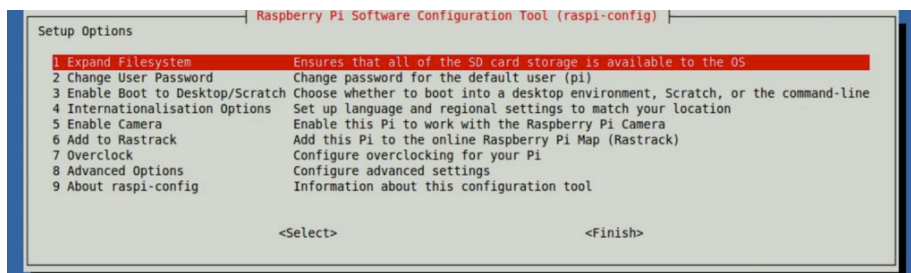
Para ello utilizamos el siguiente comando:

```
curl -sL http://ibm.biz/tjbot-bootstrap | sudo sh -
```

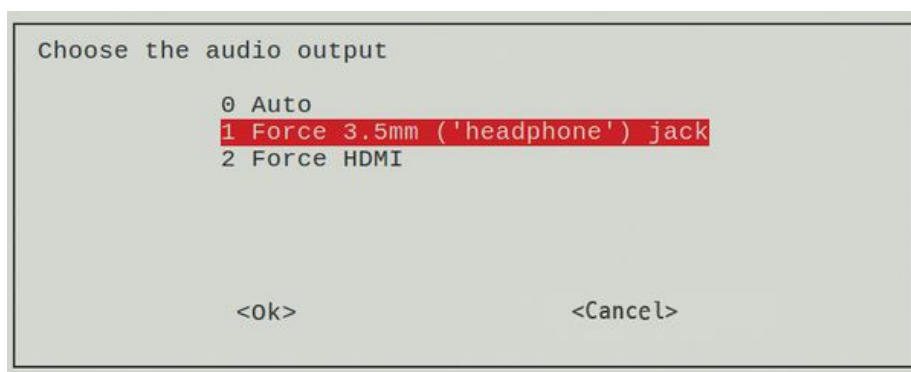
A continuación, será necesario configurar la salida de **audio**; el audio puede tener 2 opciones: **HDMI o Jack**.

Abrimos un terminal, y escribimos el siguiente comando, para abrir la configuración de las Raspberry:

```
sudo raspi-config
```



Seleccionamos "Opciones avanzadas" y luego seleccionamos "Audio". Elegimos el canal correcto para el audio de salida. Si hemos conectado un altavoz externo a la toma de audio, debemos seleccionar la toma de 3,5 mm.

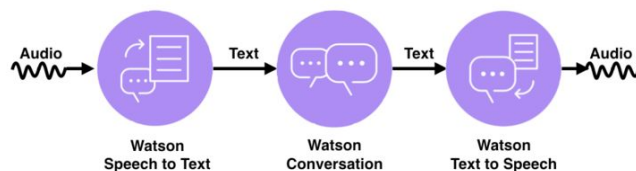


1.4.2. OBTENER EL CÓDIGO PARA TJBOT

Una vez tenemos todo configurado, necesitamos descargar el código base para el TJBot. Podemos hacerlo mediante los siguientes comandos:

```
git clone https://github.com/ibmtjbot/tjbot.git
cd tjbot/recipes/conversation
npm install
```

Una vez tenemos el código instalado, tenemos que empezar a configurar los servicios de Watson. Como ya hemos dicho, los servicios que vamos a utilizar son 3: **Text to Speech**, **Assistant** y **Speech to Text**.



Recordemos que los tres servicios estaban ya creados; ahora lo único que necesitaremos serán las credenciales para poder añadirlas al código de configuración del TJBot.

Necesitamos abrir el archivo **config.js** para actualizar dicha información. Para ello, primero hacemos una copia del archivo original (config.default.js) mediante el siguiente comando:

```
cp config.default.js config.js
```

Una vez hecho, abrimos el archivo config.js y actualizamos las credenciales de los servicios.

Será necesaria la **password**, el **username** y en el caso de Watson Assistant, el **WorkspaceID**.

```
7 // Watson Conversation
8 // https://www.ibm.com/watson/developercloud/conversation.html
9 exports.credentials.conversation = {
10     password: '',
11     username: ''
12 };
```

1.4.3. PROBANDO A TJBOT

Una vez tenemos todo configurado y la conversación creada, podemos probar a TJBot.

Abrimos el terminal y ejecutamos el siguiente comando:

```
sudo node conversation.js
```

¡Ya puedes hablar con TJBot!