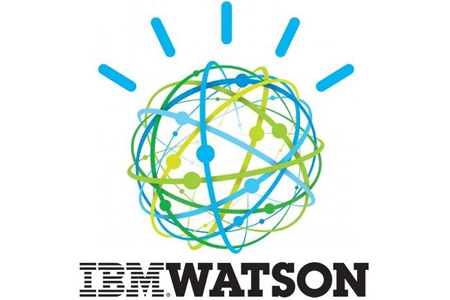
Xasdfasdfasdfasdfasdfasdasdfaskkfkkdkkdkkd



Hands-on Workshop

**IBM Developer Advocates Team**

1. Montar las apps a un repo de ibm cloud dentro de la cuenta del GEP, y poner las ligas para que lo descarguen de ahí

2. Explicación de qué servicios necesitan crear

3. Explicación de cómo conectar las credenciales

4. Ejercicio para conocer cómo funciona cada servicio (puede ser con el API Explorer o postman) en el caso de chatbot y visual pues el entrenamiento

Agenda

1. Prework
2. Try out............................................................................................................
3. Creación de servicios.....................................................................................
4. Importar workspace y configuración general.................................................
5. Configurar app nodejs con workspace...........................................................
6. Correr localmente...........................................................................................
7. Prework:

1.1 Descargar e instalar Node js

1.2 Descargar e instalar su editor de texto favorito

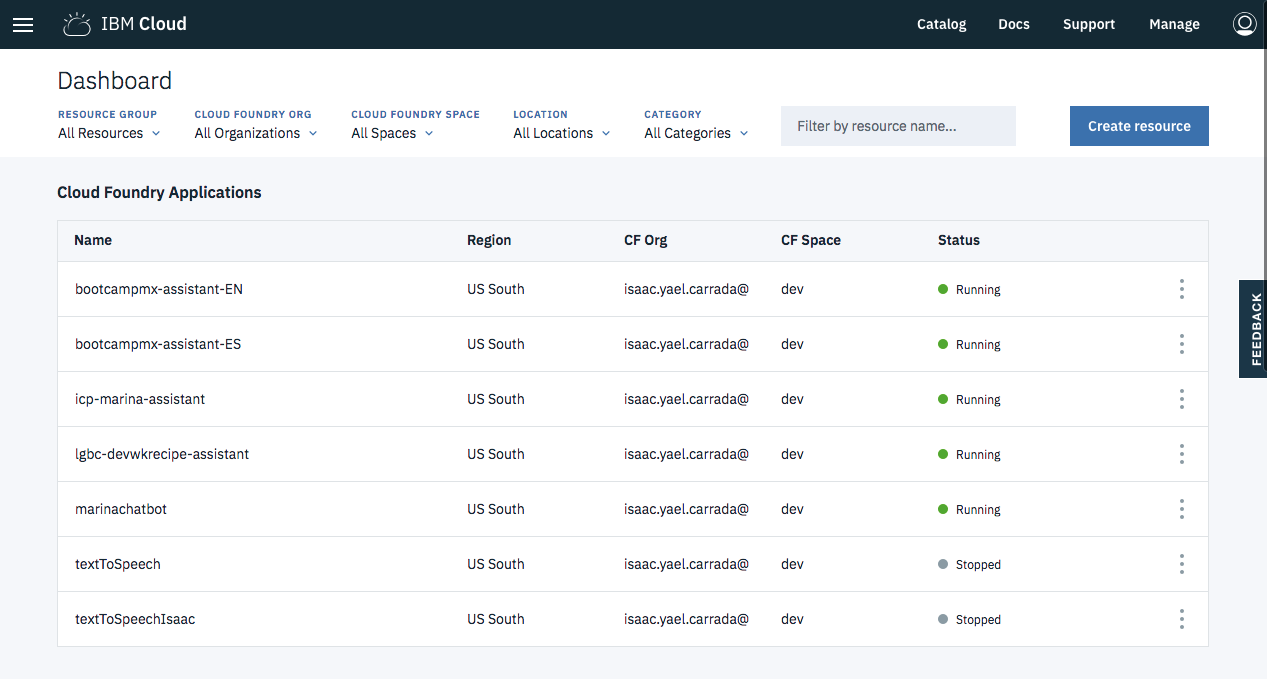
1.3 Descargar este proyecto: https://github.com/ibmdbgmx/bootcampmx-car-dashboard-es

2.1 Try out

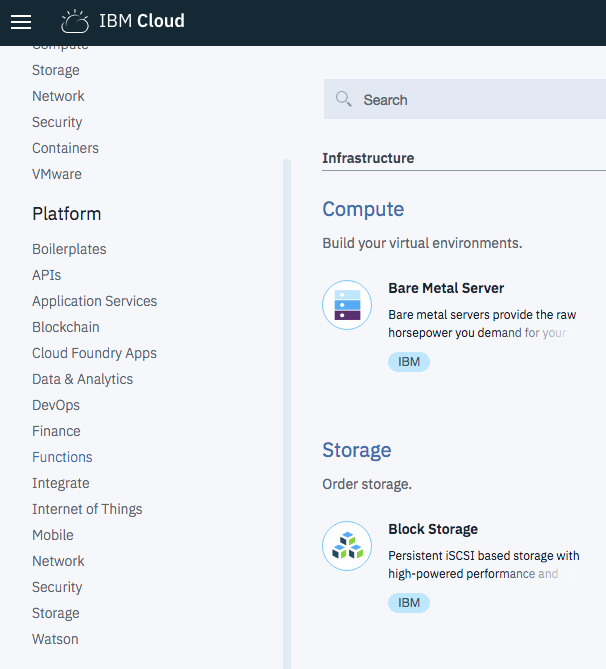
* 1. Ingresar a la URL: https://bootcampmx-assistant-en.mybluemix.net/

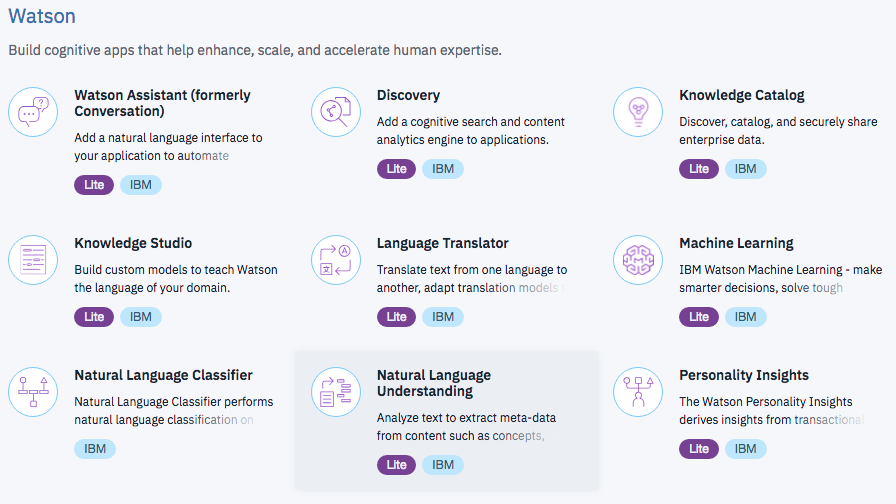
3.1 Creación de servicios.

* 1. Entrar al portal de IBM Cloud (http://console.bluemix.net)
  2. Ir al catalogo



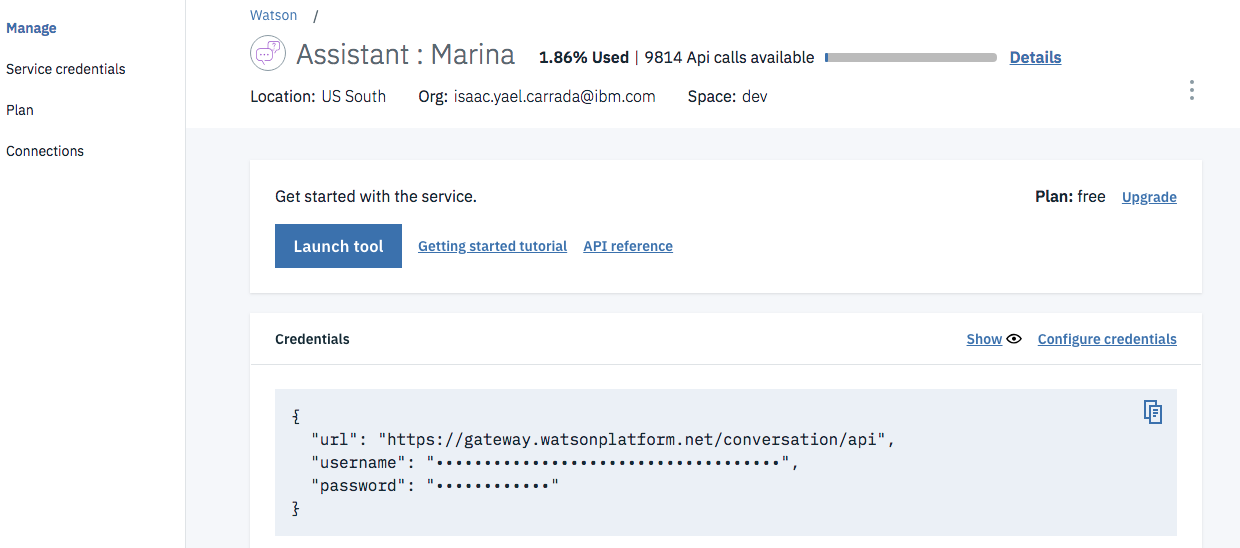
* 1. Seleccionar Watson del menú en la categoría platform

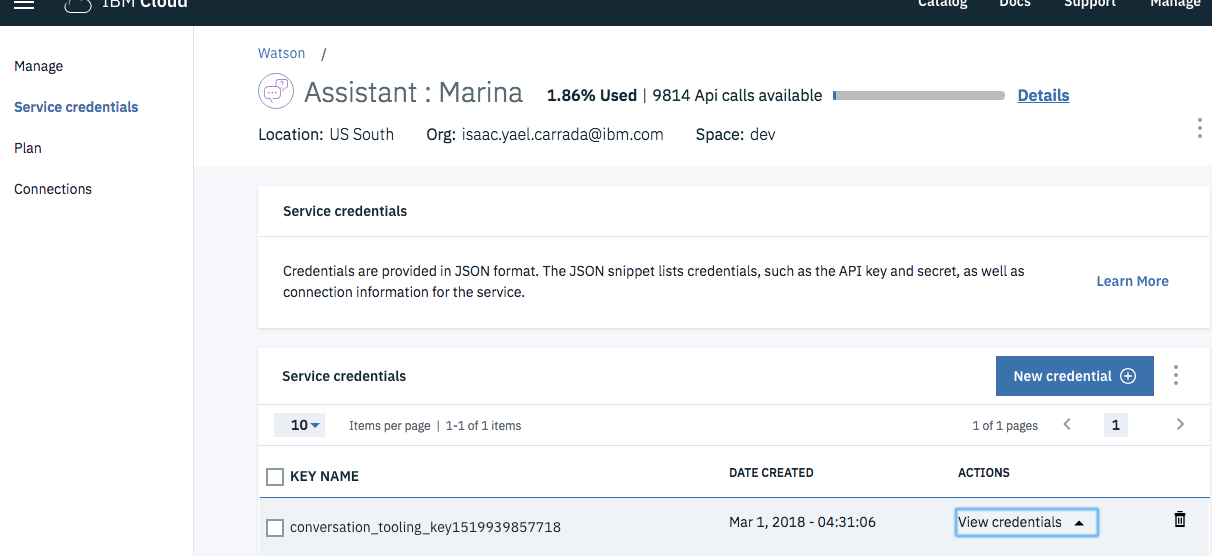


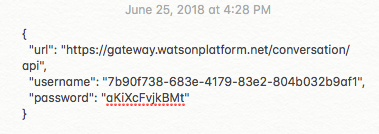
* 1. Seleccionar Assistant
  2. Nombrar app y crear servicio



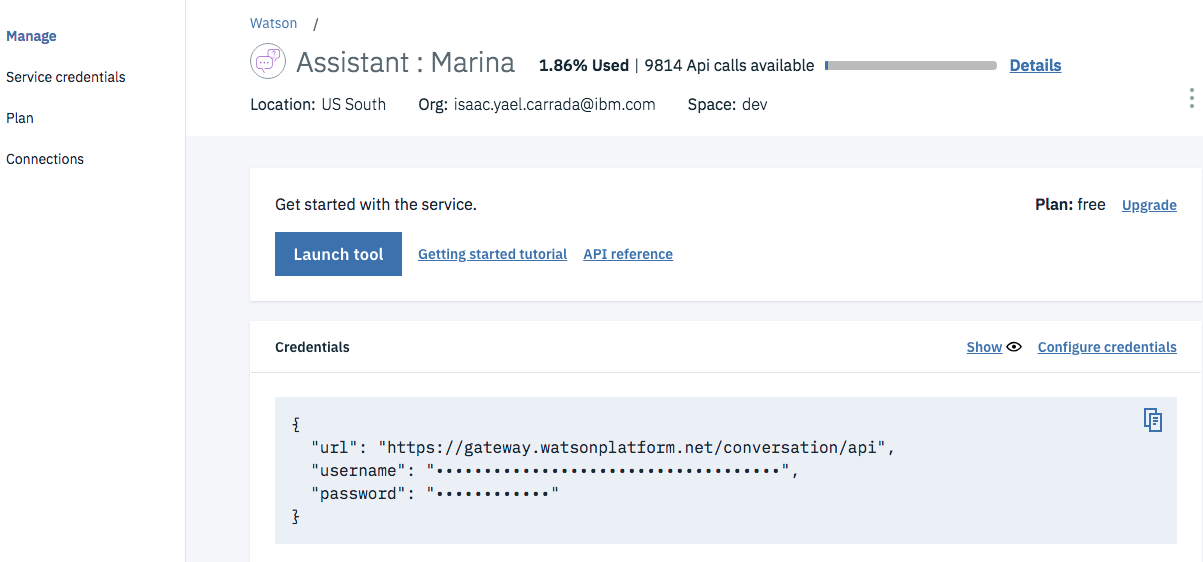
* 1. Una vez que se haya abierto el servicio nos vamos a la opcion que dice credenciales (en caso de que no nos lleve directamente al servicio una vez creado dar click en el ícono IBM Cloud)



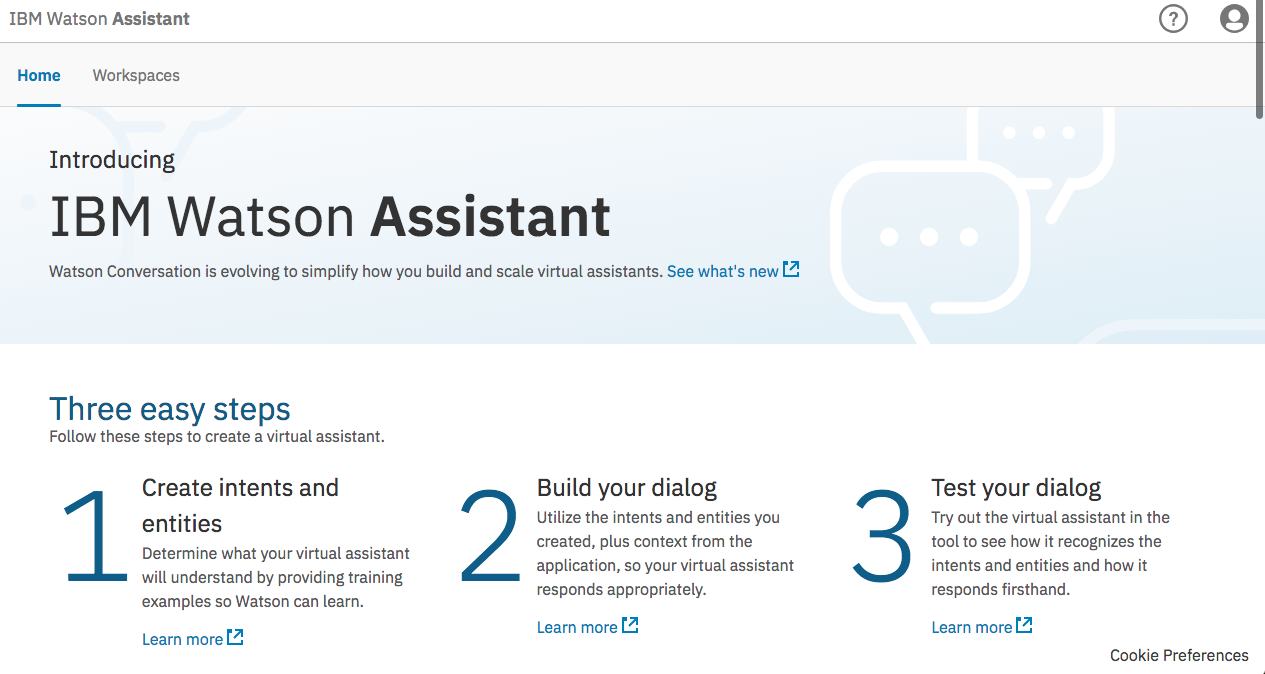
* 1. En caso de ya tener creadas credenciales hacer click en **view credentials**.En caso contrario hacer click en **New credential**
  2. Copiamos estas credenciales y las guardamos (en mi caso en Notas)



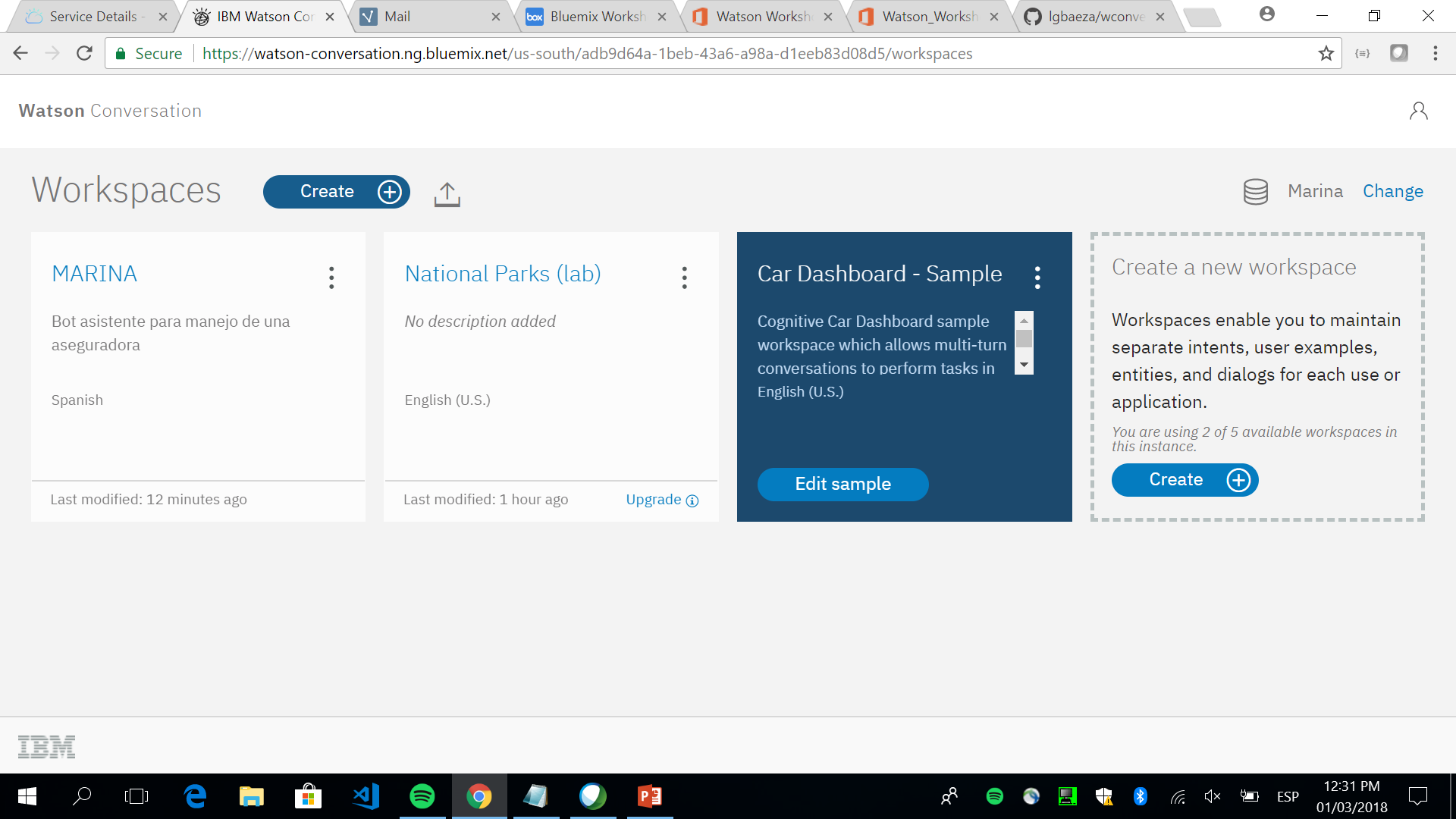
Regresamos a la pestaña que se llama Manage y damos click en **Launch Tool para** entrenar nuestro servicio



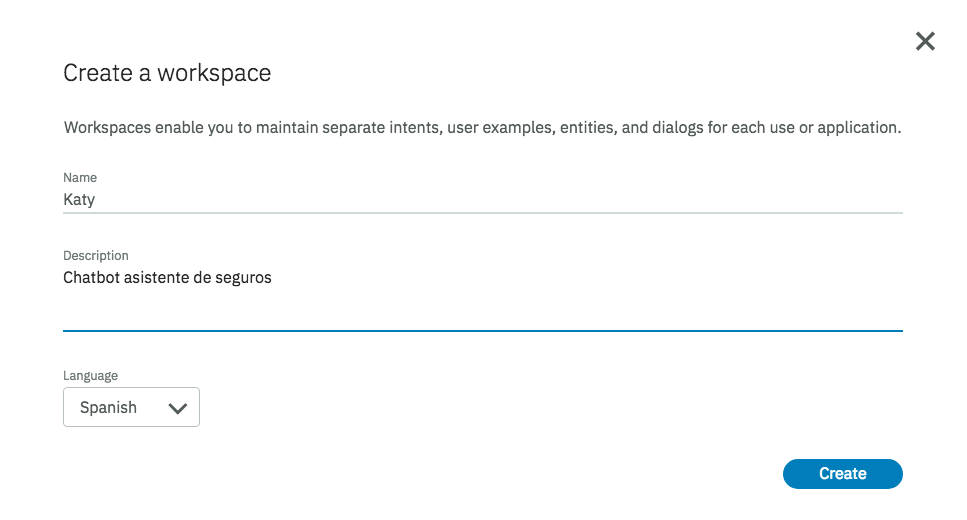
* 1. Una vez que hayamos lanzado la herramienta veremos una landing page muy parecida a esta:



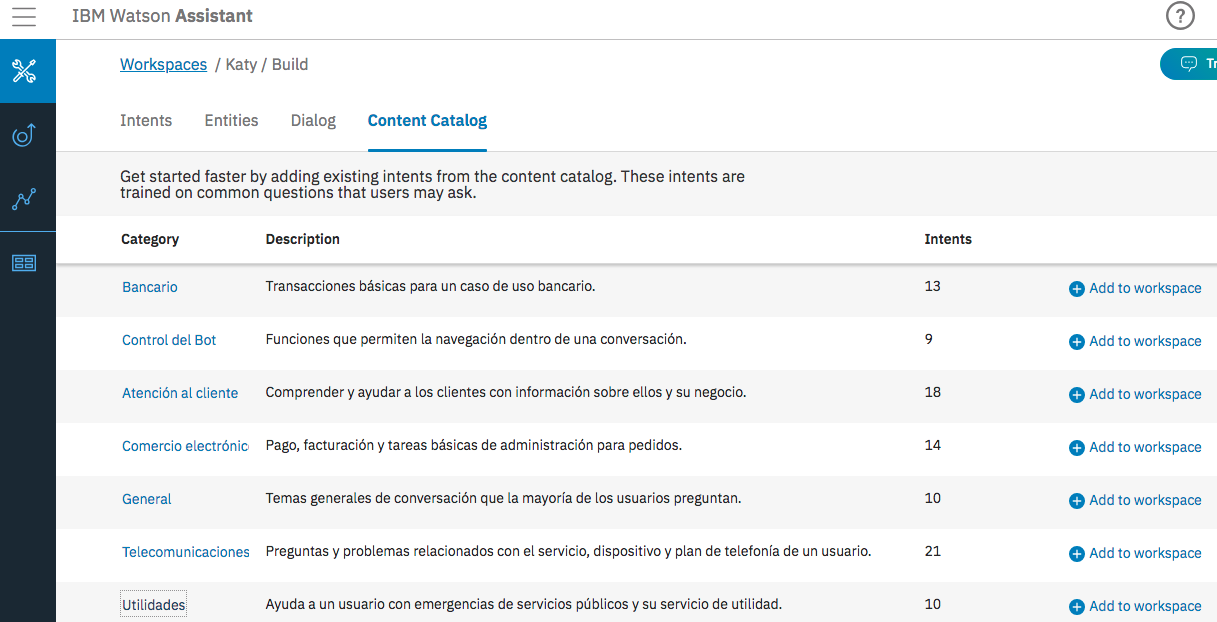
* 1. Nos movemos a la ventana llamada **Workspaces.** Cada workspace representa a una instancia de un chatbot diferente.
  2. Para crear o entrenar un chatbot utilizaremos los siguientes botones:



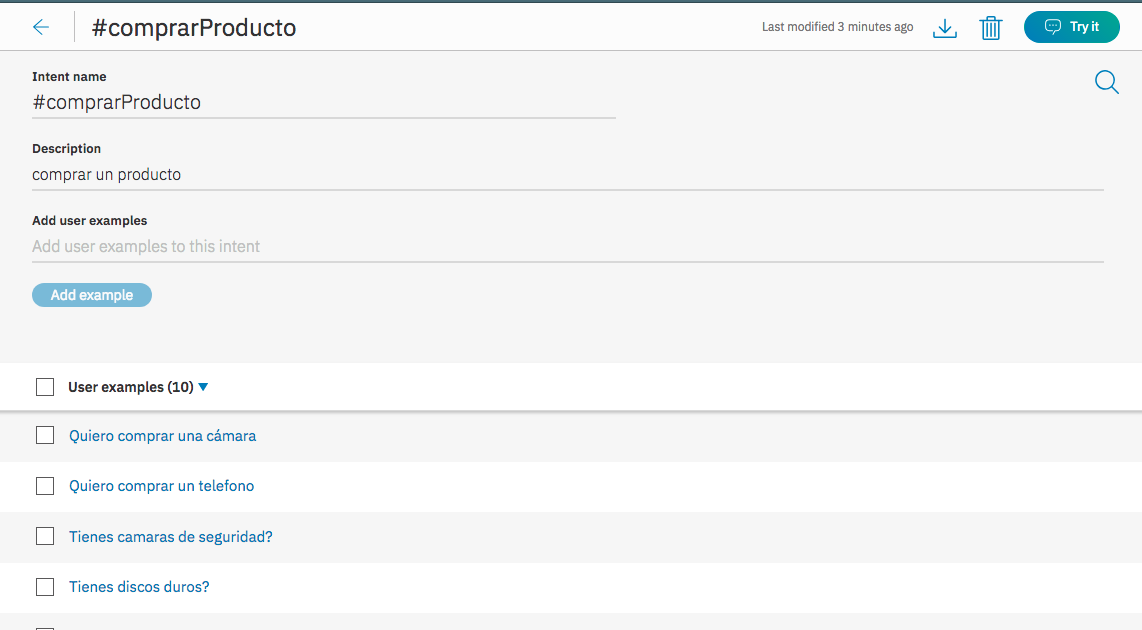
* 1. Creamos un nuevo workspace y le damos el nombre, descripción e idioma que queramos



* 1. Nos vamos a la pestaña Content Catalog

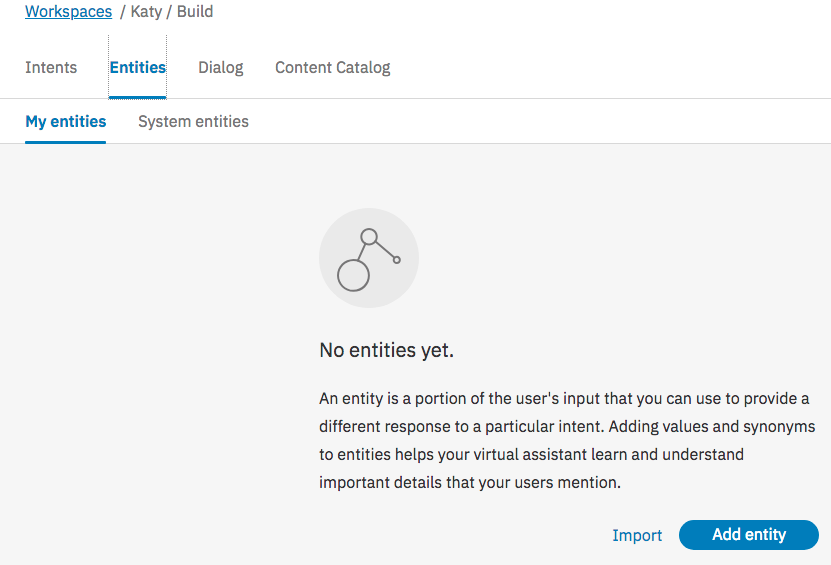


* 1. Dentro de esta pestaña podemos encontrar varios Intents ya entrenados, seleccionamos **atención al cliente** dandole click a **Add to workspace**
  2. Regresamos a los intents
  3. Navegamos por los intents para ver qué tipo de preguntas son las que podemos hacer
  4. Creamos un nuevo intent que compete un producto que querramos comprar, para esto damos click en **Add Intent**

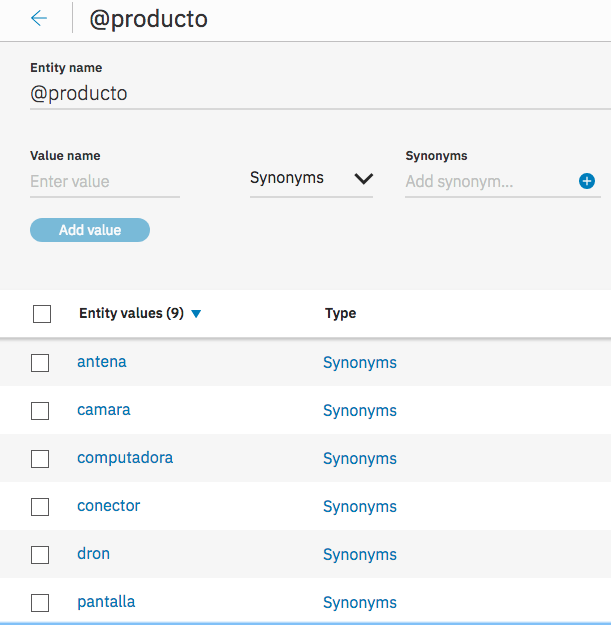


Llenamos este Intent con enunciados de cómo nuestro usuario final preguntaría respecto a nuestros productos (tienda de equipos de tecnología)

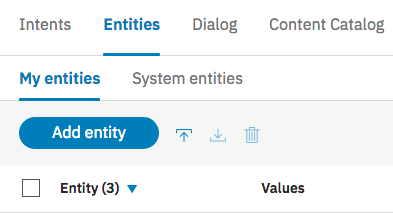
* 1. Pasamos a nuestra tab **Entities**. En esta creamos una entidad dando click en el botón **Add entity**



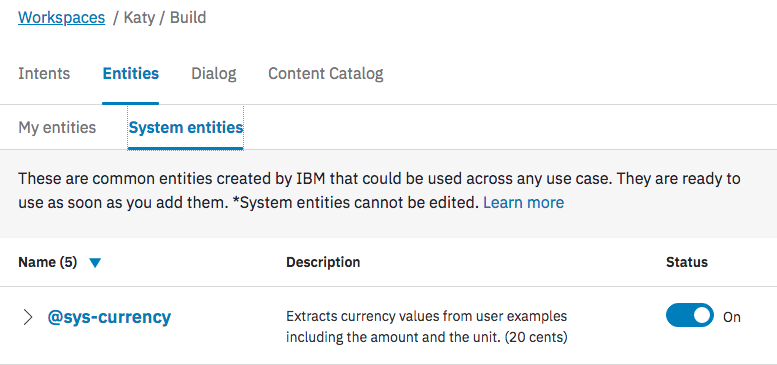
* 1. Creamos una entity para los productos que podemos comprar



* 1. De igual forma importamos los archivos que viene, en la misma carpeta training de nombre **specaudifonos.csv** y **tamañoaudifonos.csv**



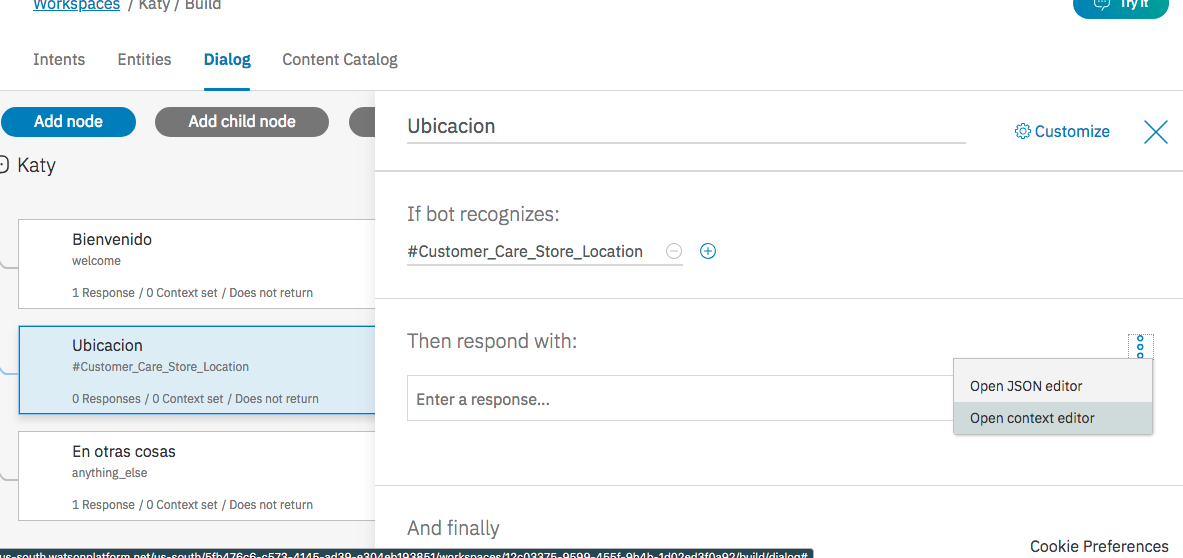
* 1. Nos vamos a system entities y activamos **@sys\_currency**



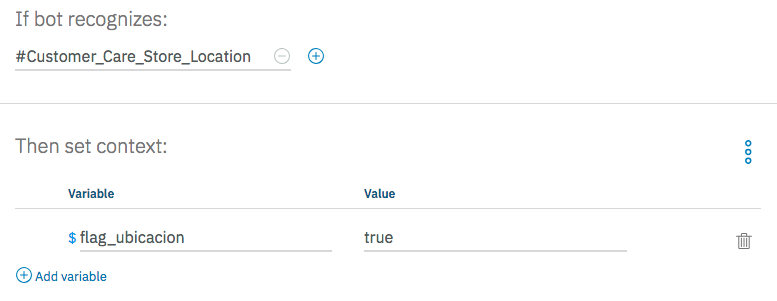
* 1. Posteriormente nos pasamos a nuestra ventana **Diálogo** y damos click en **Create**

**Dialogo:**

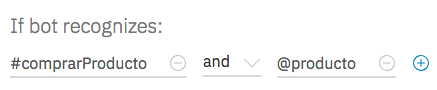
* 1. Damos click al nodo creado y le ponemos el nombre de “Ubicación”
  2. En el condicional seleccionamos el intent #Customer\_Care\_Store\_Location
  3. Abrimos el editor de contexto



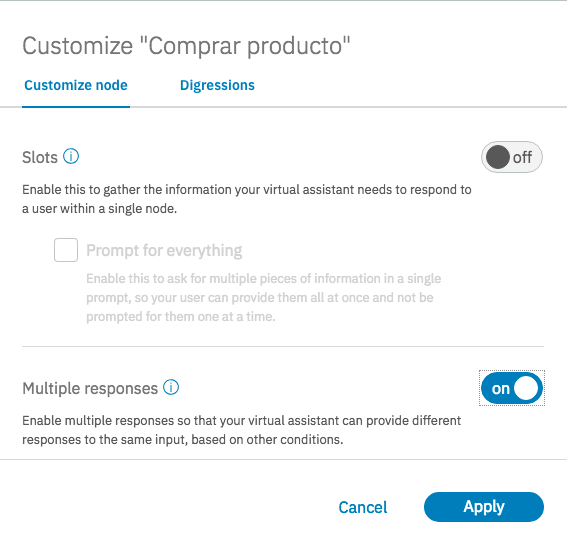
* 1. En este ponemos la variable de contexto $bandera\_ubicacion y el valor True



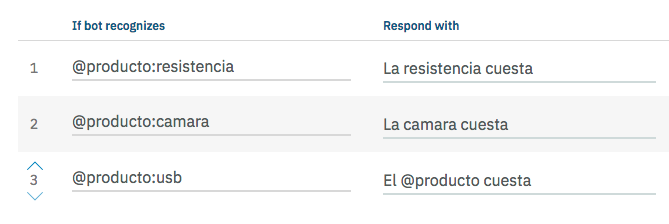
* 1. Damos una respuesta “Le envío la ubicación”
  2. Creamos un nodo más, esta vez que se llame Comprar producto
  3. Seleccionamos el intent comprarProducto y la entidad producto



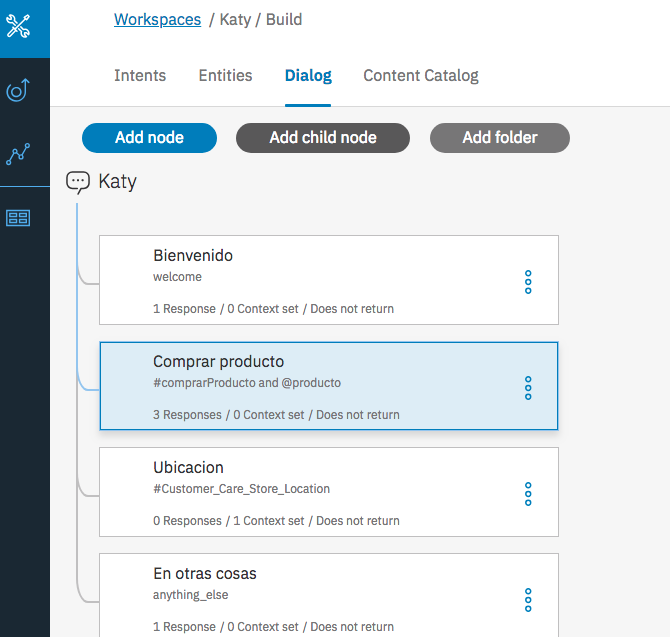
* 1. Seleccionamos el boton customize y activamos **Multiple Responses** y damos click en Apply



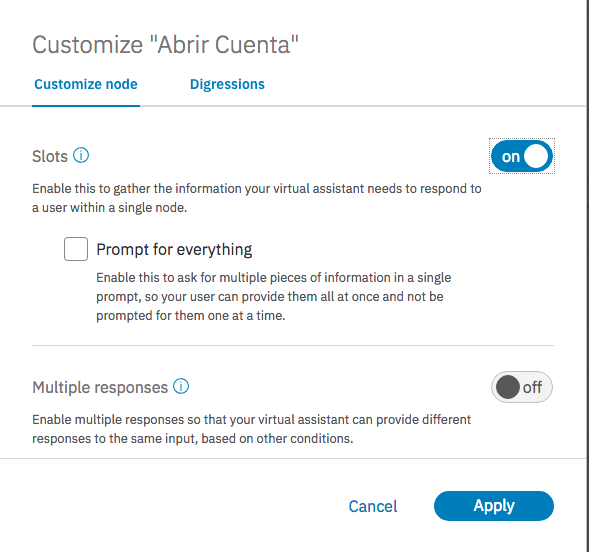
* 1. De lado izquierdo seleccionamos producto, colocamos un “**:”** y después de esto seleccionamos a lo que tiene que ser igual el producto:



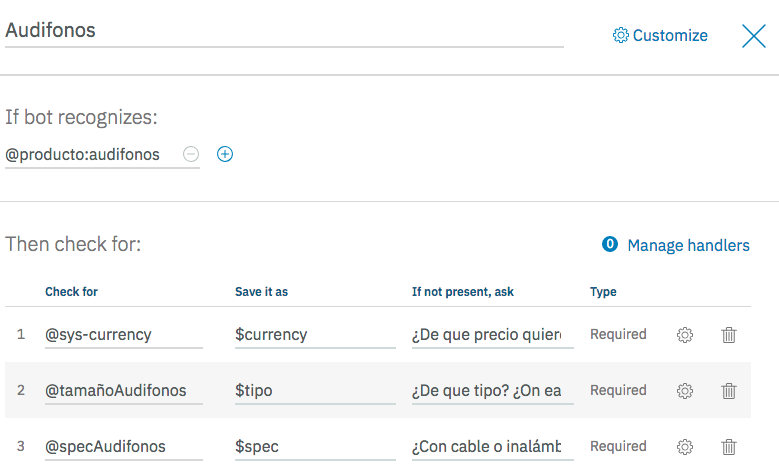
* 1. Añadimos un nodo hijo de este nodo, de la siguiente forma:



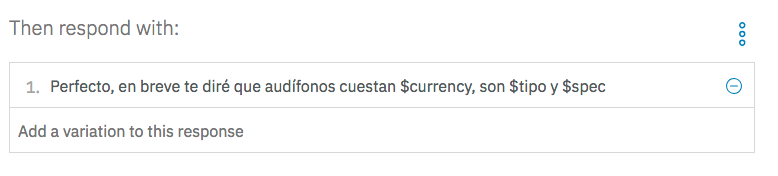
* 1. Dentro de este nuevo nodo le ponemos el nombre @**producto:audifonos**. De la misma forma poniendo como condición que se haya seleccionado la entidad producto con el valor de cámara
  2. Seleccionamos customizar y seleccionamos **slots**



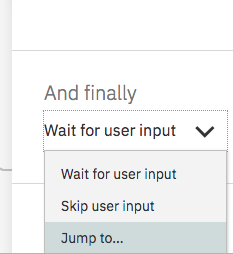
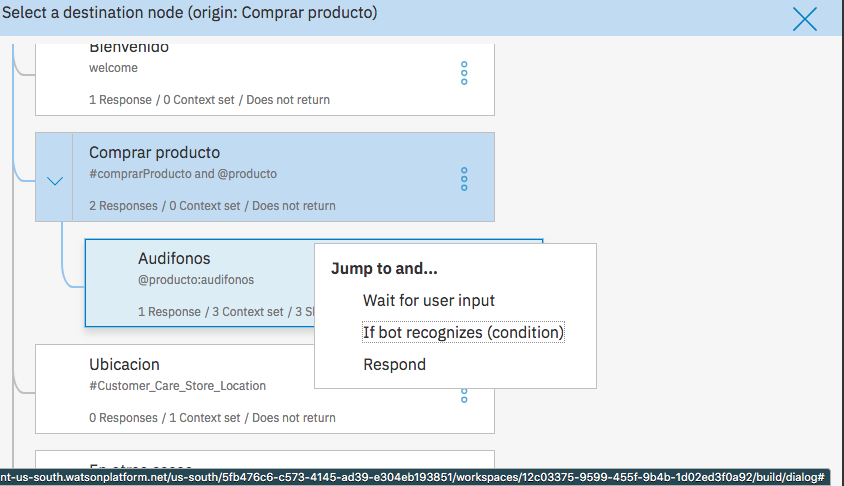
* 1. Una vez que ya tenemos la funcionalidad de los slots los llenamos de la siguiente manera:



* 1. Para la respuesta:



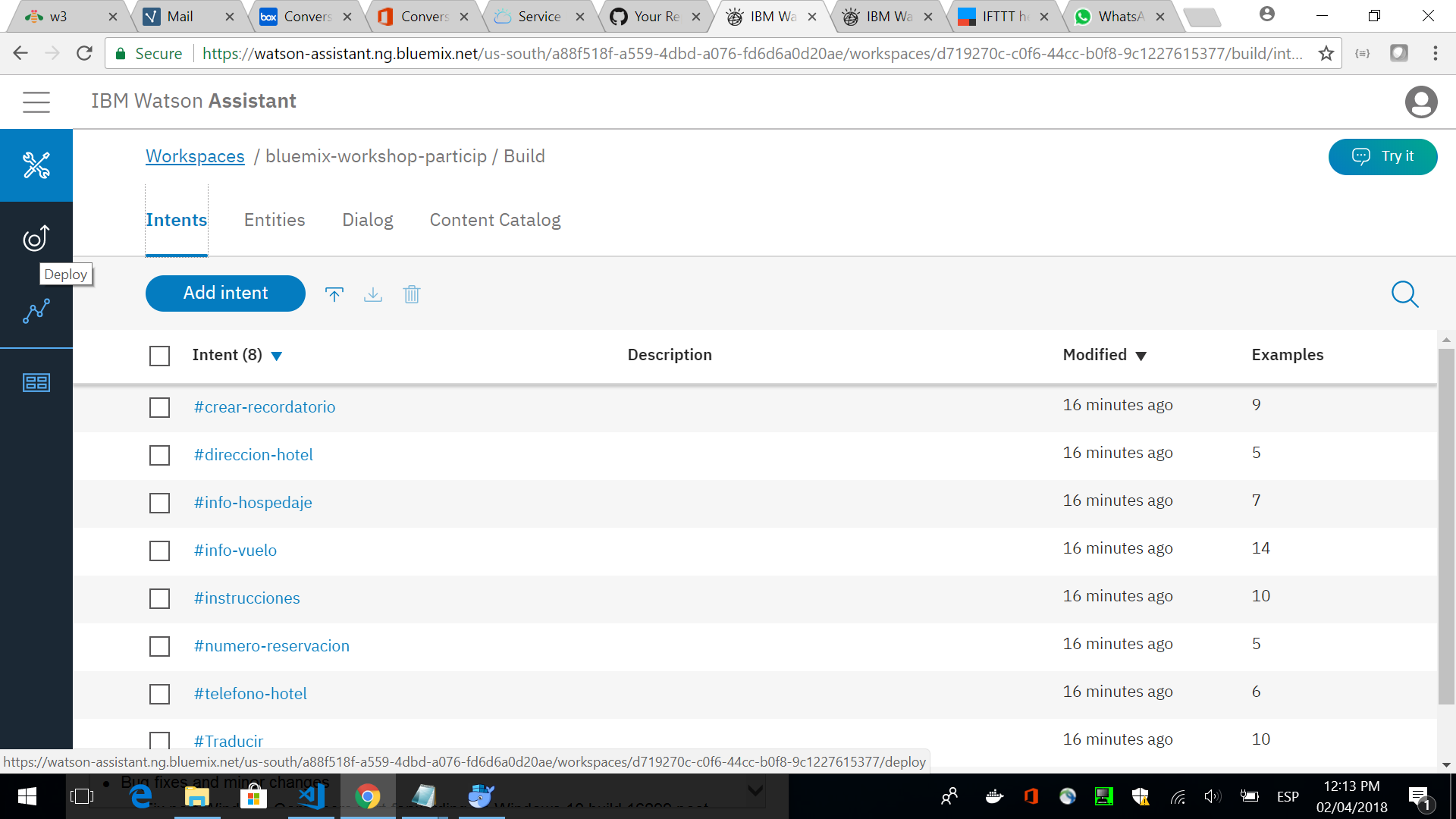
* 1. Una vez que tengamos este nodo nos vamos al padre (comprar producto) y cambiamos la respuesta de **Wait for user input** a **Jump to** y seleccionamos el hijo y después **If bot recognizes condition:**

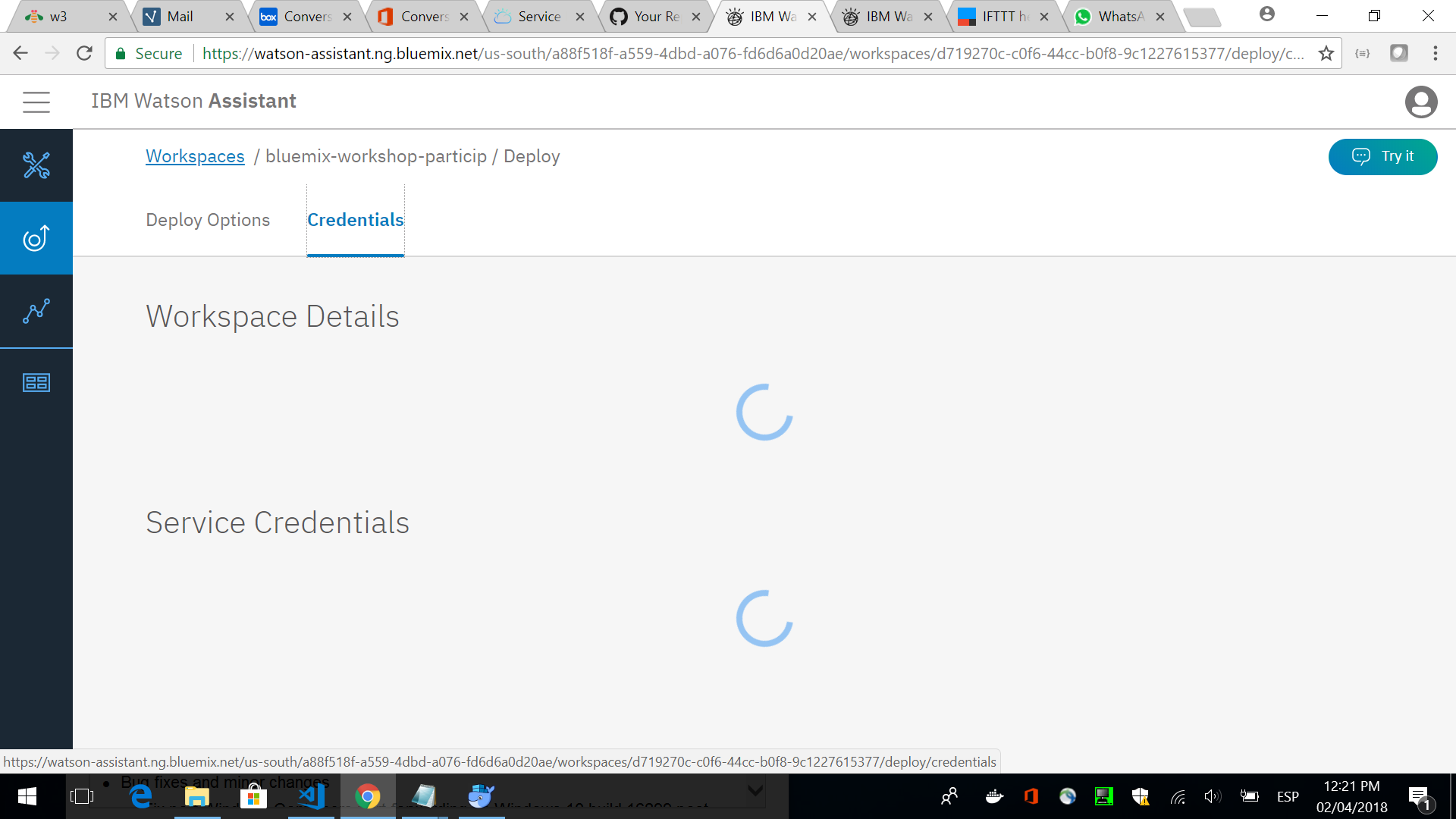
* 1. Una vez hecho esto ya podemos pasar a la siguiente sección

**4 Configurar app nodejs con workspace**

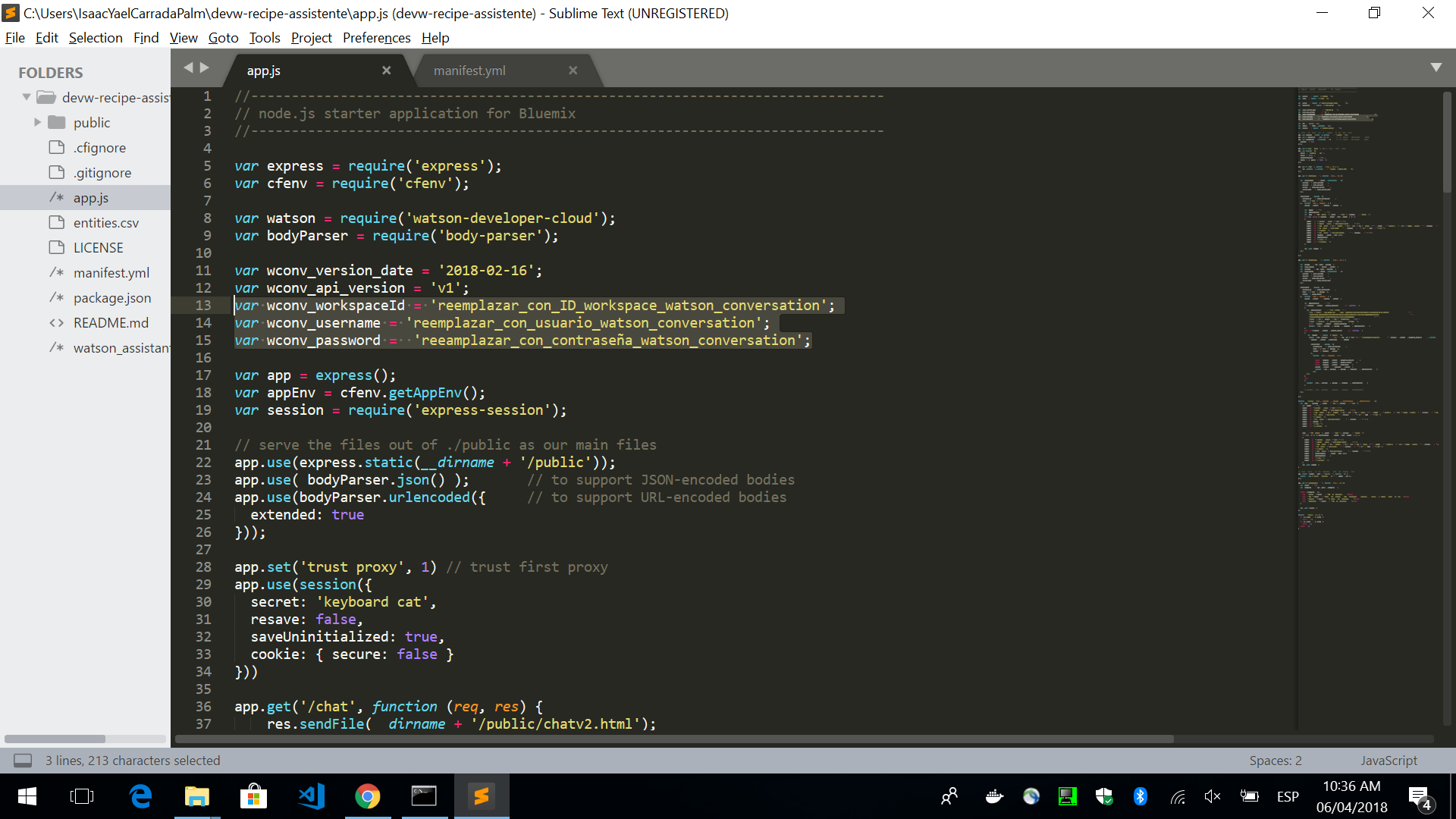
1. Hacer click en el ícono Deploy



1. Seleccionar Credentials



1. Aquí podemos encontrar el WorkspaceID, Username y Password, que utilizaremos en el siguiente paso
2. En el archivo app.js, para las variables workspaceID, username y password utilizar estos valores

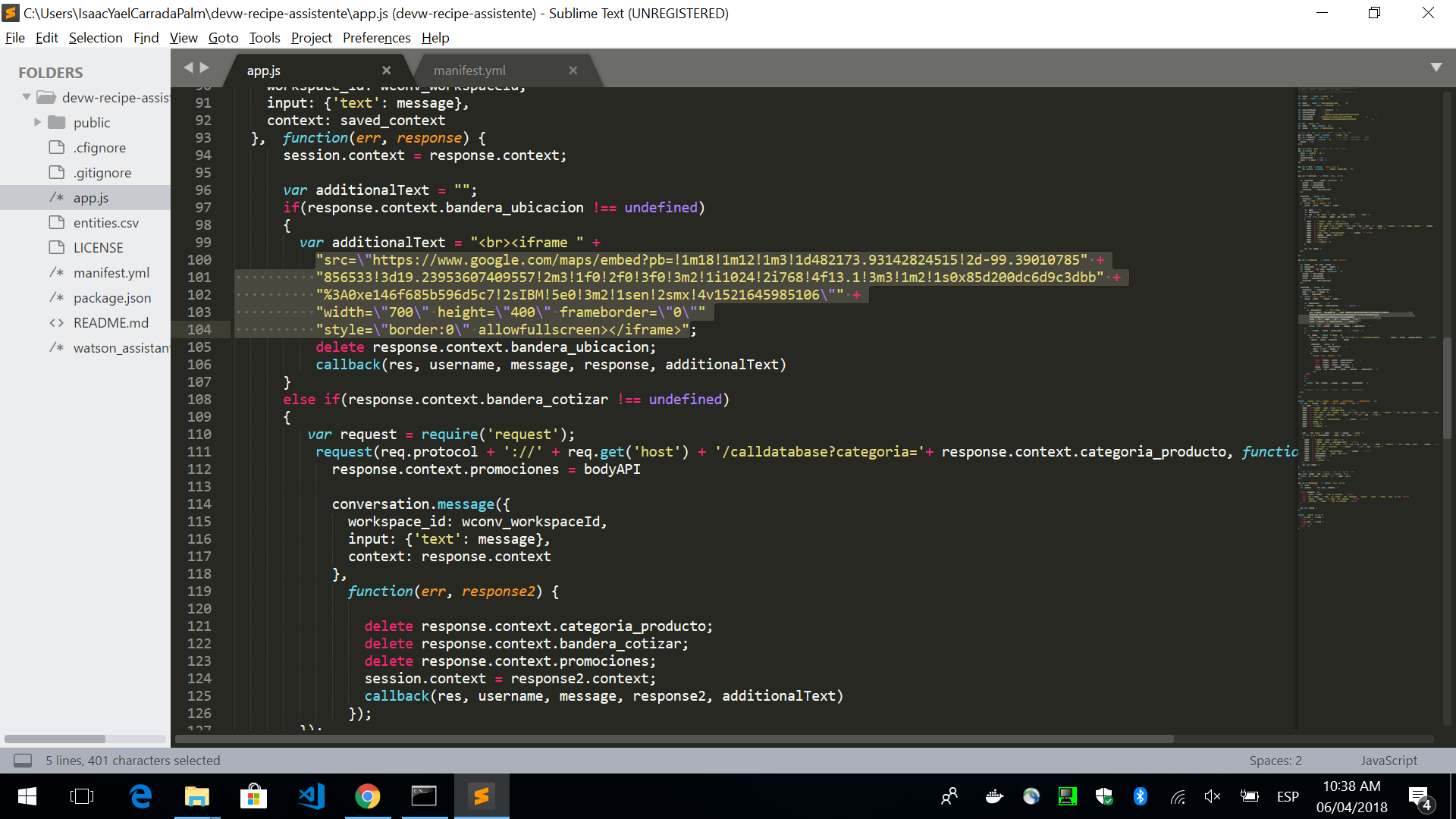


5.5 Correr localmente

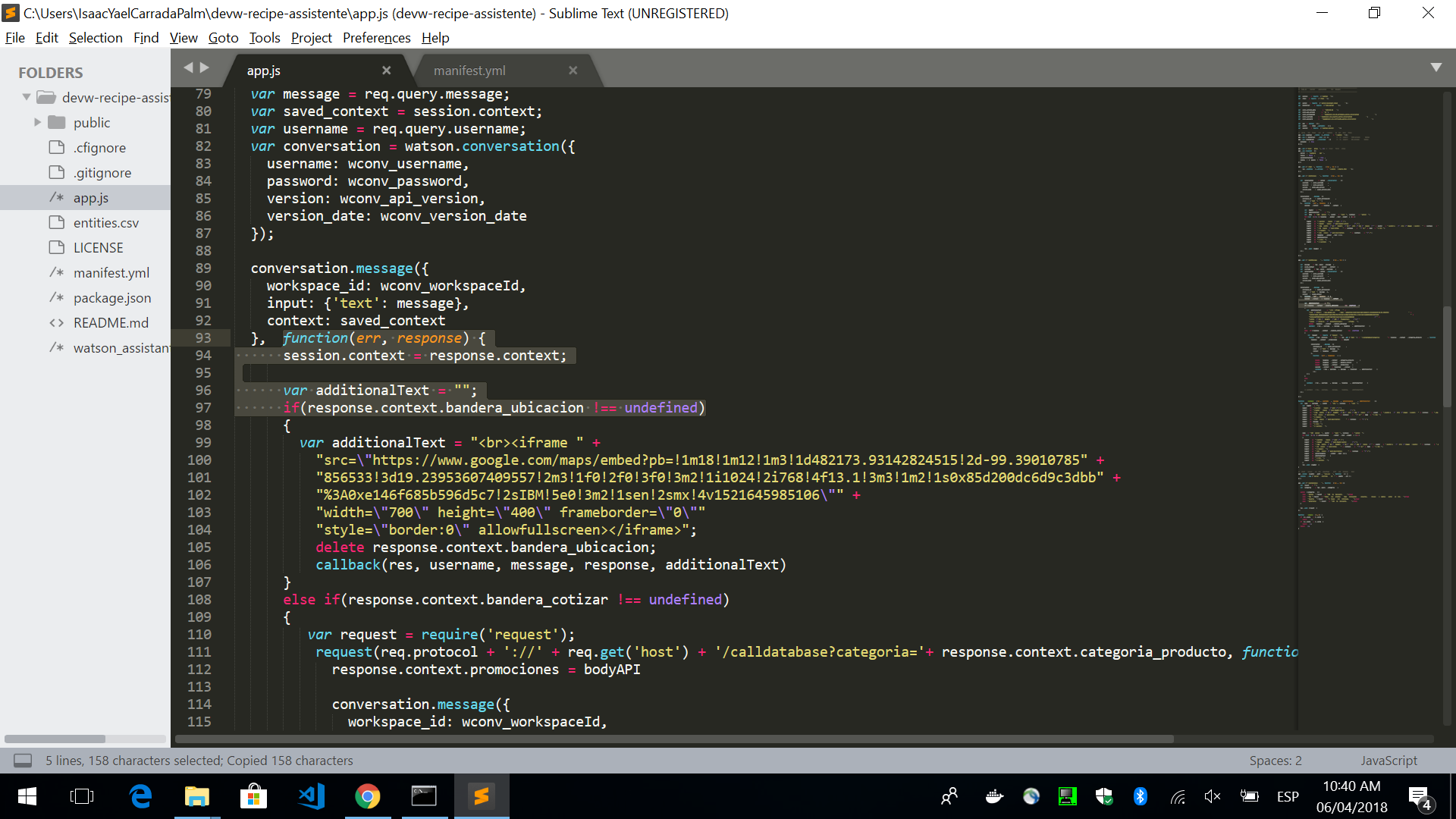
1. Abrir la línea de comandos
2. Ir al directorio en el que se encuentra nuestro proyecto
3. Ingresar: npm install
4. Ingresar: node "app.js"
5. Corre en localhost

5.6 Añadir un nuevo intent y dialog

1. La parte del código en la que está configurada la dirección del mapa es:



1. La función de respuesta con la bandera ubicación se encuentra definida en la siguiente parte



5.7 Correr localmente de nuevo para notar el cambio