

Sistema de Recomendaciones con Azure Machine Learning y Bot Framework

Introducción

El uso de Bots, con la integración de herramientas como Machine Learning, permite acercar a los usuarios al negocio, funcionando como facilitador en el acceso a la información de negocio al ofrecer la mejor solución a problemas del mundo real.

Los Bots se encuentran prácticamente en todas partes, si sabes dónde buscar, o al menos así será en los próximos años. A pesar de ser Bot que interactúan con los usuarios, ello no implica que sean Bots dentro del contexto de la inteligencia artificial. Surge así la pregunta ¿Qué factores determinan que un Bot sea inteligente?

Machine Learning y Procesamiento de Lenguaje Natural son los pilares en la definición de Bots inteligentes, ¿qué tanto impacta esta Inteligencia Artificial a la percepción de los usuarios cuando platican con los Bots? Un aprendizaje valioso que han dejado herramientas de Asistente Personal como Cortana es que los usuarios puedan hacer preguntas a las cuales el Bot responda con información relevante y con esto facilitarles encontrar valor en la conversación.

Las herramientas actuales de Machine Learning permiten aprender de las charlas que los Bots entablan con los usuarios, en la medida que otorgan la habilidad a los Bots para sensor, pensar y actuar. A través de Machine Learning los Bots “escuchan” y son capaces de “entender” lo que los usuarios quieren o lo que están buscando para tareas como obtener información o realizar una acción, de esta forma el Bot toma decisiones basadas en el ambiente de la plática.

Por su parte, el Bot necesita un grado de confianza basado en la información que aprende sobre los usuarios y el lenguaje utilizado, de tal manera que la interpretación permita dar al usuario la mejor respuesta posible. Un Bot existe es aquel que tiene una plataforma inteligente con Machine Learning y que está diseñado para dar solución a problemas del mundo real.

El futuro de los Bots inteligentes está lleno de un sinnúmero de posibilidades, sin embargo, el factor determinante que hace resaltar a un Bot de los demás es su habilidad para recolectar y almacenar información valiosa de los usuarios, y así poder tomar decisiones basadas en el análisis de datos otorgando respuestas relevantes de forma rápida y efectiva.

Objetivo: Construir un Bot de recomendaciones usando Microsoft Bot Framework que se integra con un modelo de recomendación en Azure Machine Learning.

Prerrequisitos

Lo siguiente es requerido para poder realizar el laboratorio:

- Una suscripción activa de Microsoft Azure
- Conocimientos básicos de programación

Índice

Ejercicio 1: Crear los recursos de Bot Framework en Azure

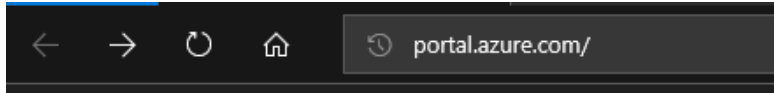
Ejercicio 2: Exploración del Web IDE

Ejercicio 3: Configuración del Bot

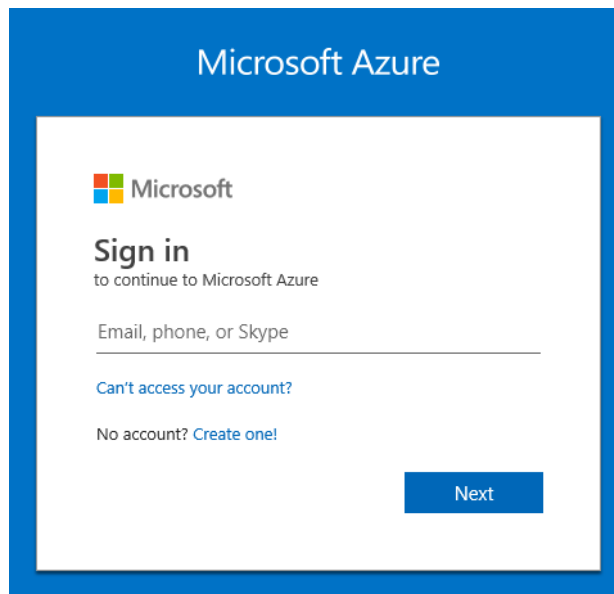
Ejercicio 4: Publicar el Bot en la página web

Ejercicio 1: Crear los recursos de Bot Framework en Azure

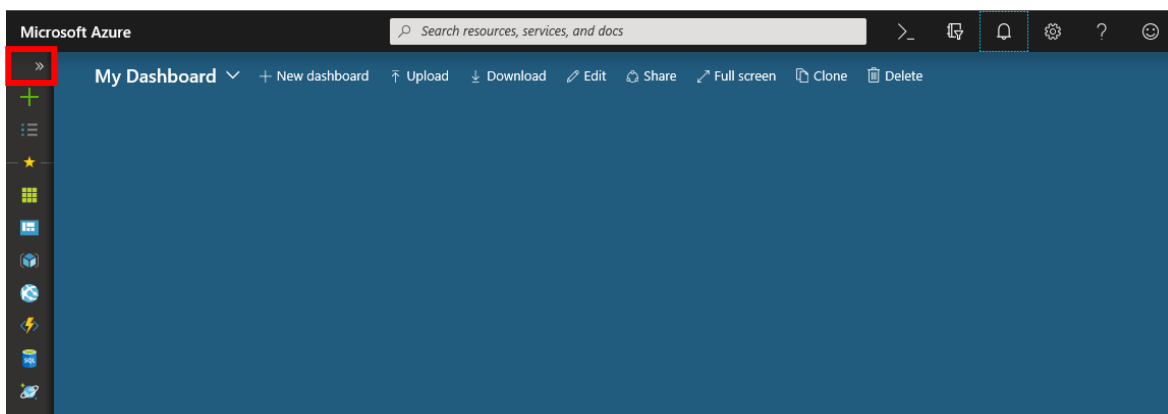
1. Abrir la página del [Portal de Azure](https://portal.azure.com/).



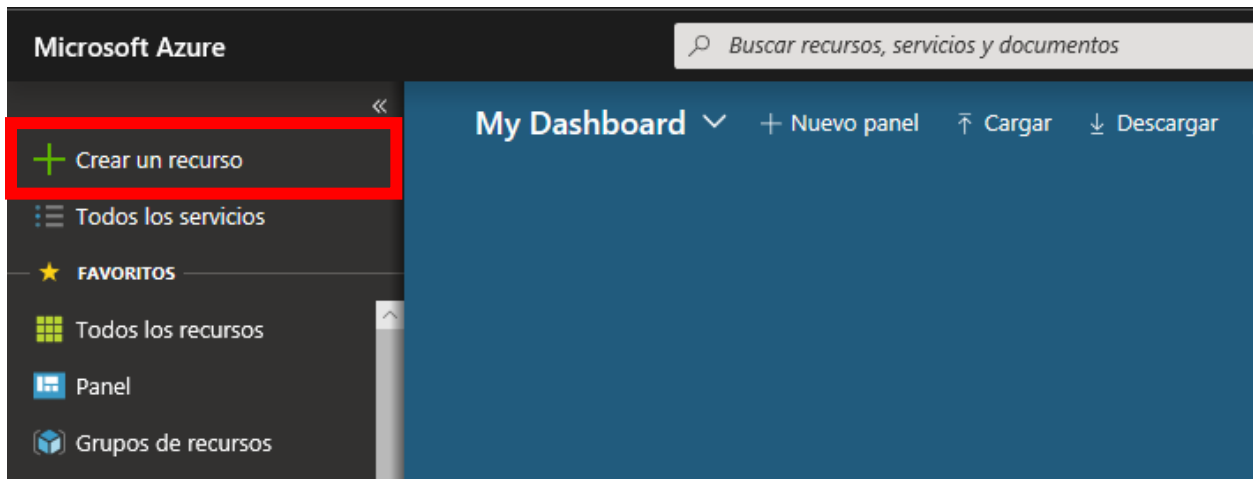
2. Acceder con la cuenta de correo registrada en con su Azure Pass



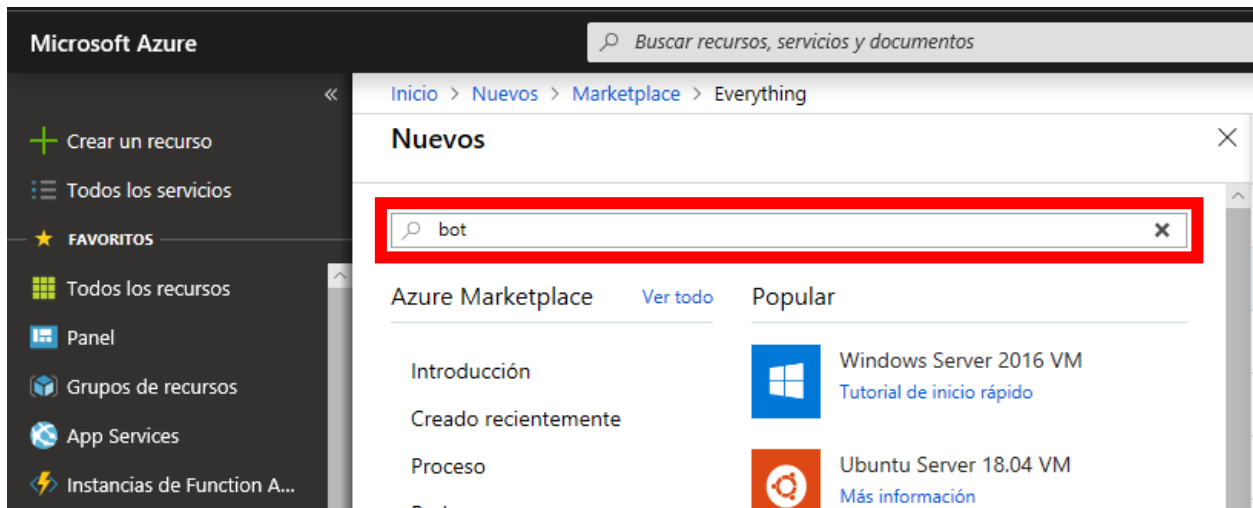
3. Esta es la página de inicio del **Portal de Azure**. Hacer clic en la flecha doble de la barra izquierda para desplegar el menú de opciones en caso de estar colapsado.



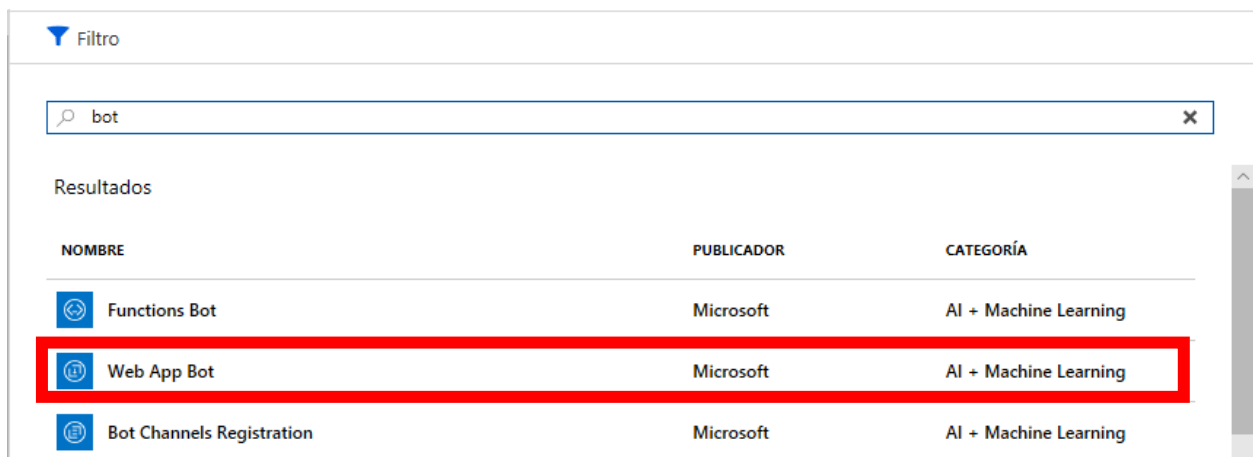
4. Hacer clic en "Crear un Recurso".



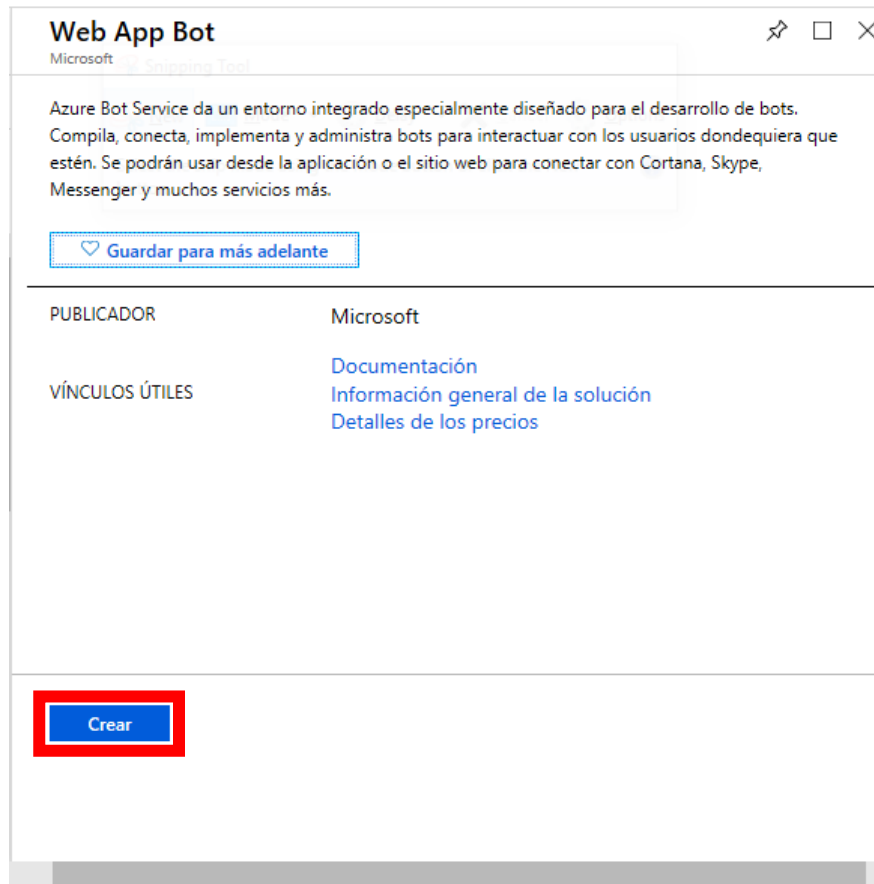
5. Buscar la palabra "bot" en la barra de búsqueda.



6. Dar clic en "Enter" para que despliegue las opciones de Bot y **seleccionar "Web App Bot"**.



7. Se desplegará una ventana en la que se describe este recurso y hacemos **click en el botón "Crear"**.



8. Se cargará una ventana con una forma en la que cargamos los siguientes datos:
- a. Nombre del bot ([iniciales de tu nombre]_bot).

* Nombre del bot ⓘ

- b. Suscripción
- c. Grupo de recursos

* Grupo de recursos

- d. Ubicación (Ubicaciones en EEUU para mejor rendimiento)
- e. Plan de tarifa (F0 (10K mensajes premium))
- f. Nombre de la aplicación (debe ser una cadena de caracteres única, ya que el servicio genere una URL pública para hacer la conexión con el servicio de Bot).

* Nombre de la aplicación ⓘ

aaabot ✓

.azurewebsites.net

- g. Plantilla de Bot (Basic Bot (C#))
 - i. Seleccionar SDK versión 3.0
- h. Plan de App Service/Ubicación

Web App Bot > Bot de aplicación web > Plan de App Service

Plan de App Service

Seleccionar un plan para la aplicación web

+ Crear nuevo

Nuevo plan de App Service

Crear un plan para la aplicación web

* Nombre del plan de App Service

botSP ✓

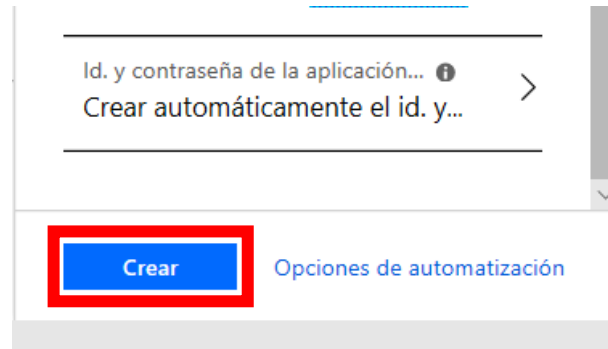
* Ubicación

Centro de EE. UU. ▼

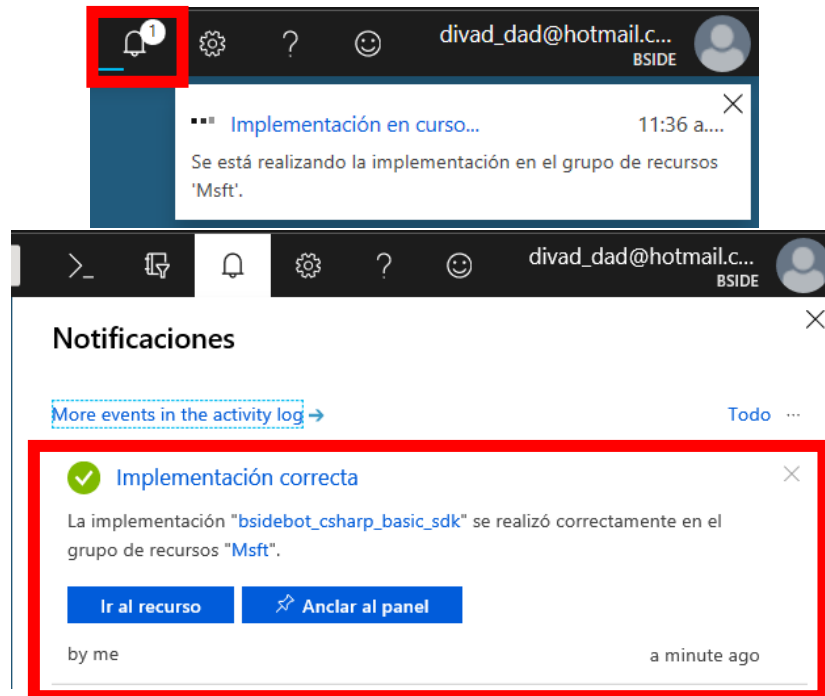
Actualmente, el plan de tarifa tiene el val...
predeterminado "S1 Standard". Puedes
modificarlo si visitas la [página de recursos del plan de App Service](#) una vez hayas
creado el recurso, o también puedes elegir
un plan existente en la suscripción.

Aceptar

- i. Azure Storage (Crear nuevo, autocompletado de forma automática).
- j. Application Insights (Desactivado)
- k. Id y contraseña de la aplicación (Se genera de forma automática)
- l. Hacer clic en crear

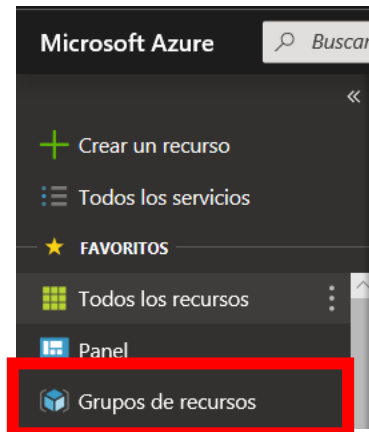


9. Una vez validados los campos de la forma inicia el proceso de creación de los recursos.

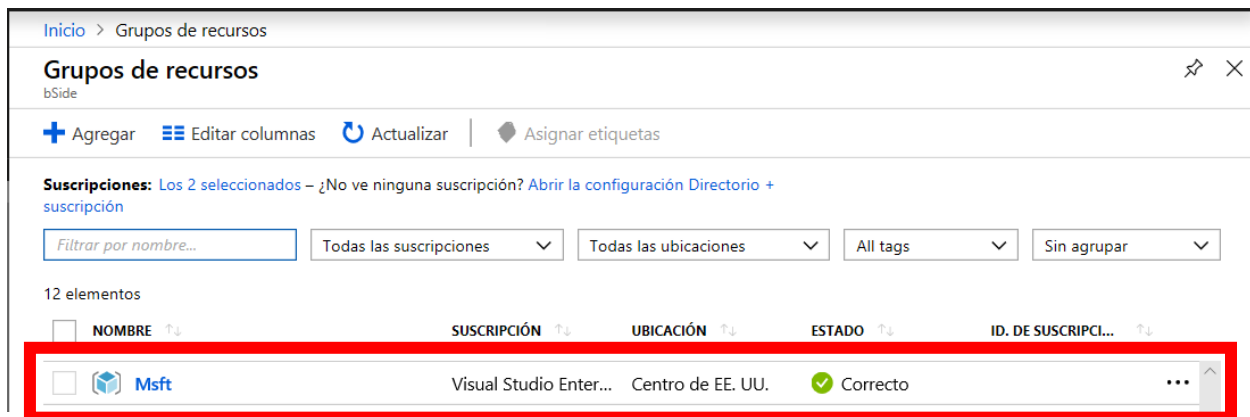


Ejercicio 2: Exploración del Web IDE

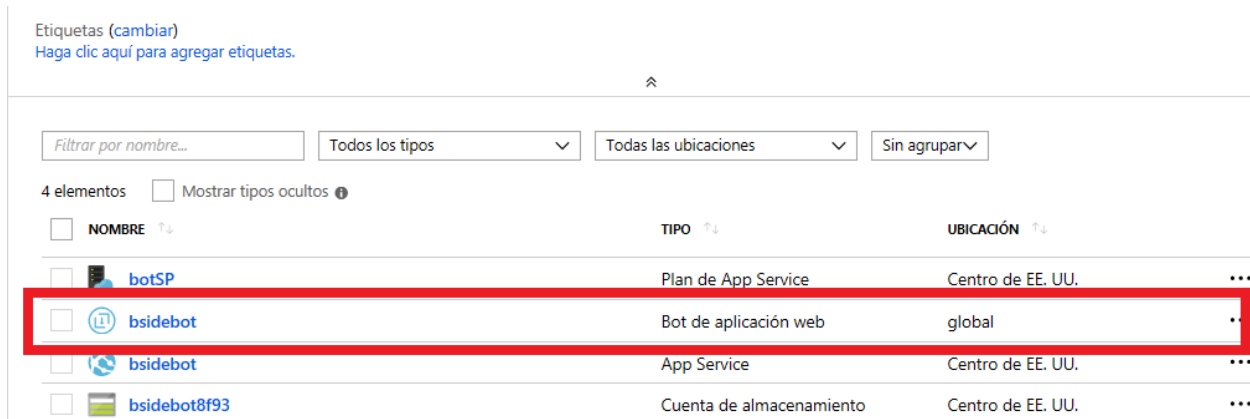
1. En la página principal del Portal de Azure, **ir a los Grupos de recursos** creado.



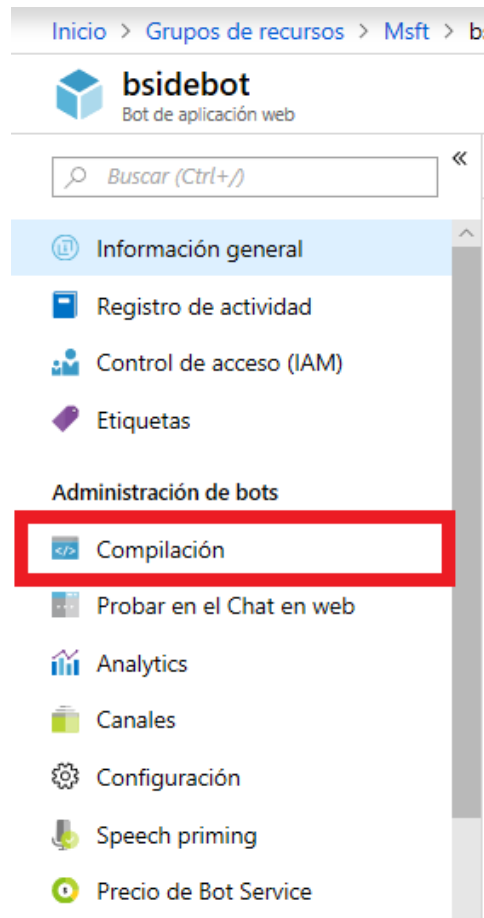
2. **Seleccionar el Grupos de recursos** creado.



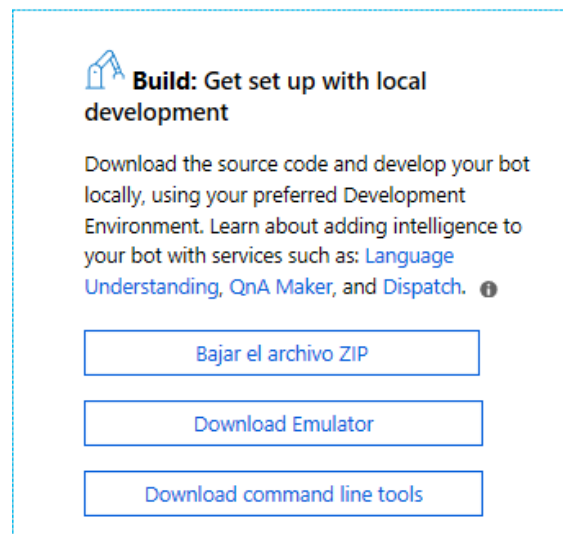
3. Dentro del Grupo de recursos se muestran los recursos creados para una solución de Bot Web. **Seleccionar** el recurso con el Tipo "Bot de aplicación web".



4. En el panel del recurso, en la sección de “Administración del bot”, **seleccionar “Compilación”**.



5. Existen tres opciones al momento de seleccionar la opción de compilación:



a. **Build**

- i. **Bajar el archivo ZIP.** Con esta opción se descarga el código del Bot en formato ZIP. Este archivo contiene el proyecto de Visual Studio con las clases y archivos de configuración necesarios para iniciar con el desarrollo.
- ii. **Download Emulator.** Abre un link de descarga del ejecutable de instalación de la aplicación Bot Framework Emulator, la cual es útil cuando se tiene un ambiente de desarrollo local y se desea probar el Bot.
- iii. **Download command line tools.** Esta opción provee un link de descarga de comandos de Azure que permiten manipular recursos de nube y Bots en aplicaciones web.



Test: Use emulator to test and refine

Connect to services, refine and debug locally with [Emulator](#). Test your bot online in [Web Chat](#).

b. **Test**

- i. Dentro del **Portal de Azure**, se utiliza el Web chat para realizar las pruebas de funcionalidad, se revisará el Web Chat más adelante.



Publish: Upload to Azure

When you are ready, publish your bot to Azure and [connect it to Channels](#).

Learn about publishing from local development [directly to Azure](#)

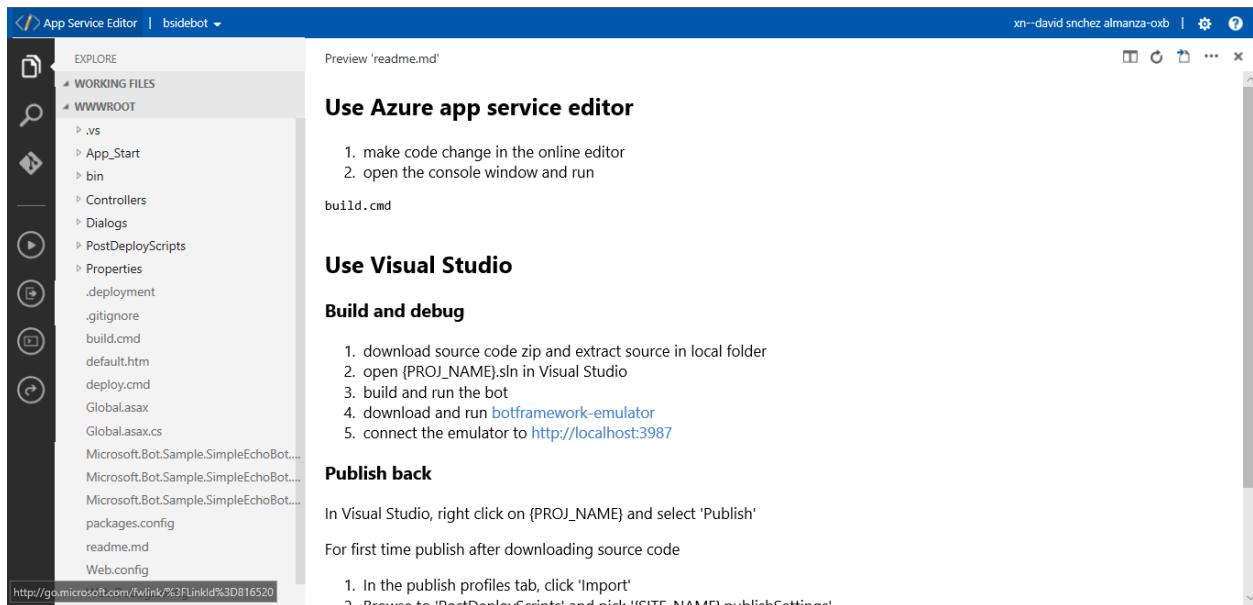
Publish updates automatically to Azure with [Continuous deployment](#)

c. **Publish**

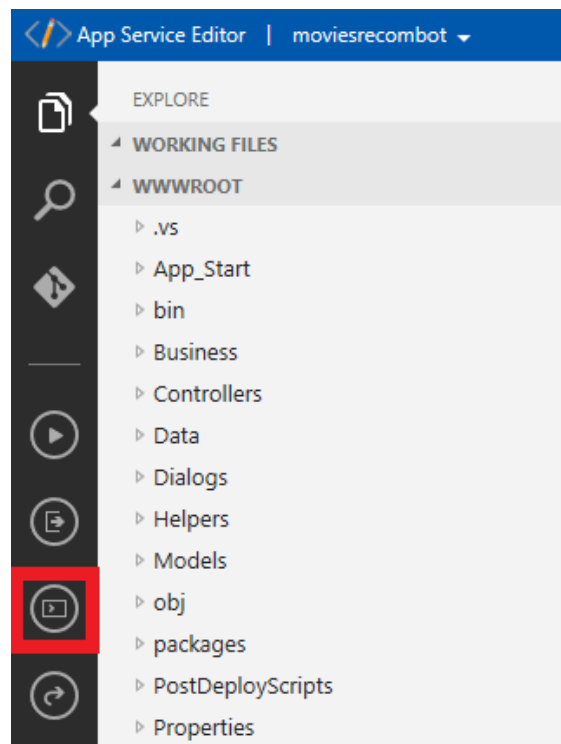
- i. El Bot se puede publicar desde Azure a canales con distintos proveedores de aplicaciones de chat.
 - ii. La plataforma de Bot Framework permite desplegar directamente en Azure
 - iii. La integración con despliegue continuo permite definir la ubicación del código fuente que se utilizará para hacer los despliegues (p. ej. Github).
6. Seleccionar la opción de "Abrir el editor de código en línea" que se encuentra en la esquina inferior derecha.

Cambie rápidamente el código del bot en línea, ejecute build.cmd en la consola del editor y visualice los cambios inmediatamente [Abrir el editor de código en línea](#)

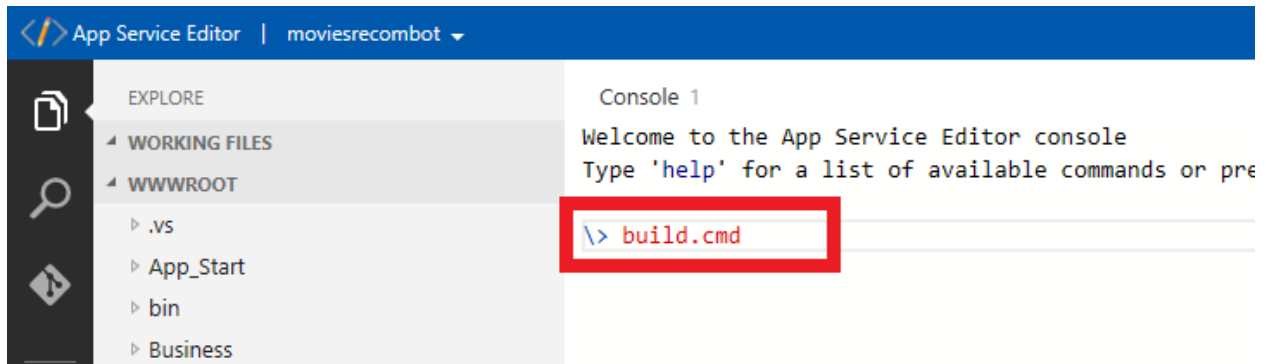
7. Al abrir el link el navegador muestra la siguiente pantalla:



8. Se hará una compilación de prueba para validar el directorio de la solución. Para ello, sobre la barra de la izquierda, seleccionar el tercer ícono del segundo grupo de íconos.

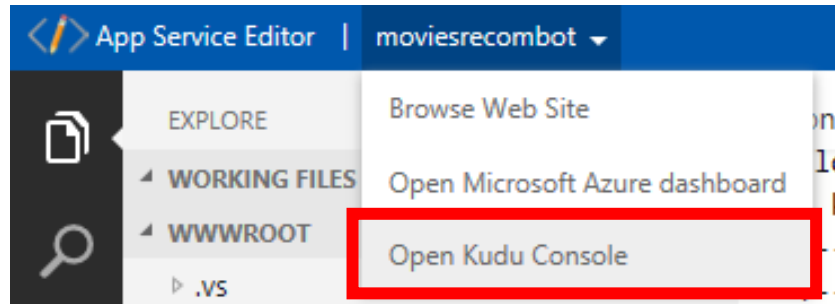
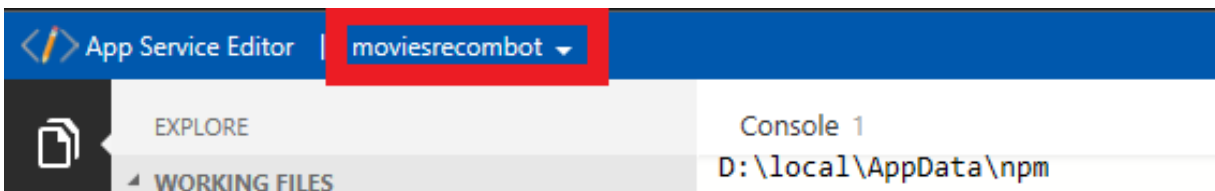


9. En la consola web, escribir el comando `build.cmd`. Este comando instalará Kudu Sync, el cual administrará la instalación de paqueterías del bot.

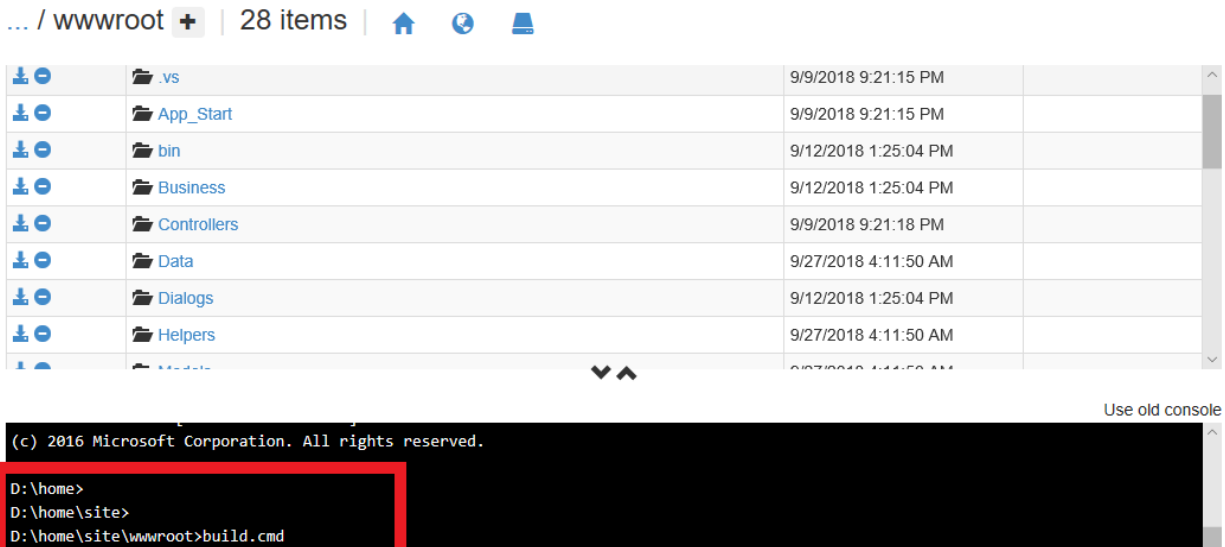


10. Si el comando devuelve errores, se deben seguir los siguientes puntos, de otra forma es posible ir al punto 11.

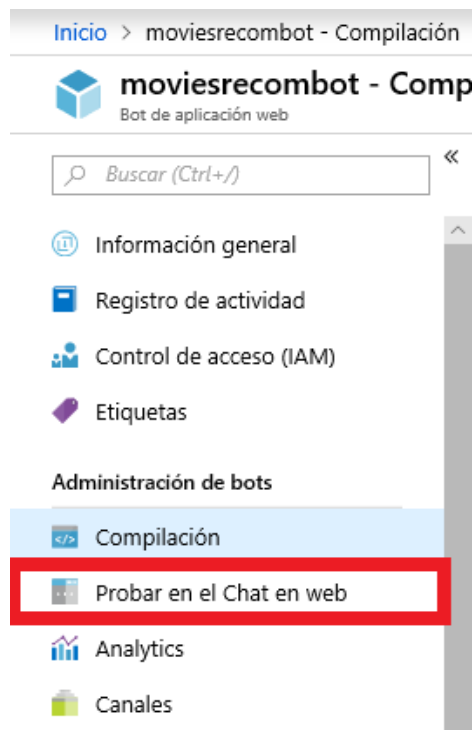
- a. Selección el combo con el nombre del Bot en la esquina superior izquierda, y seleccionar la opción **Open Kudu Console**.



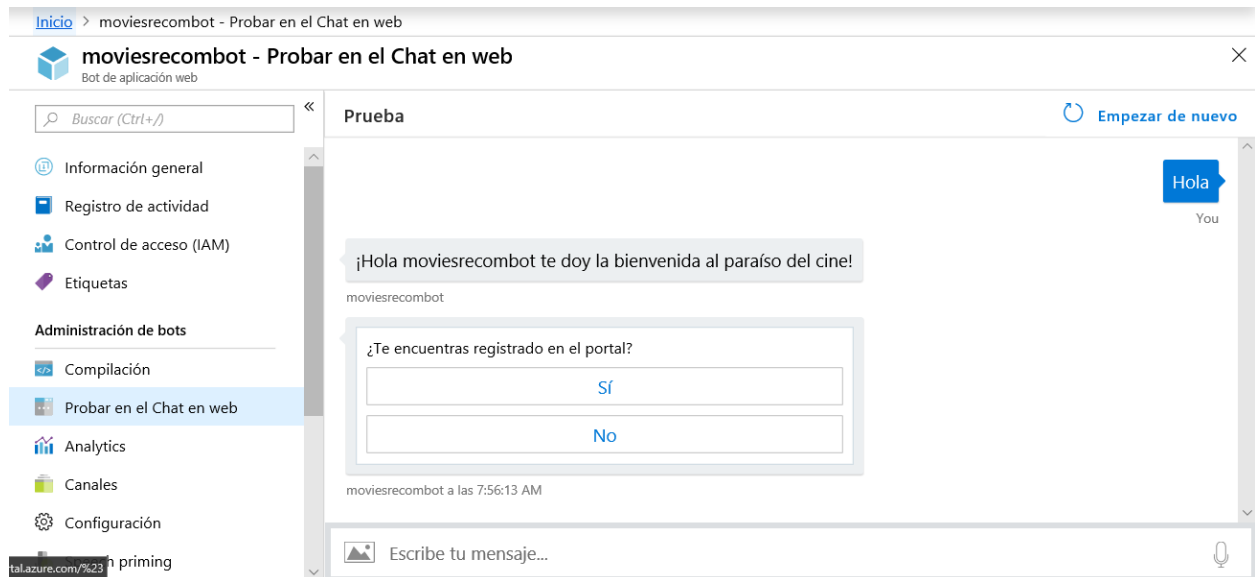
- b. En la consola de Kudu, navegar al directorio: D:\home\site\wwwroot>
Desde este directorio se debe correr el comando `build.cmd`



11. El servicio de Bot Framework ofrece una interfaz de Web Chat embebida para realizar pruebas, sin embargo, la recomendación es contar un ambiente de desarrollo y pruebas local que permita el mejor uso de las herramientas disponibles como [Visual Studio CE](#) y [Bot Framework Emulator](#). Para fines del Workshop, el IDE web es más que suficiente.
12. El Web chat de pruebas se localiza dentro de las herramientas de administración de Bots "Probar en el chat en web".



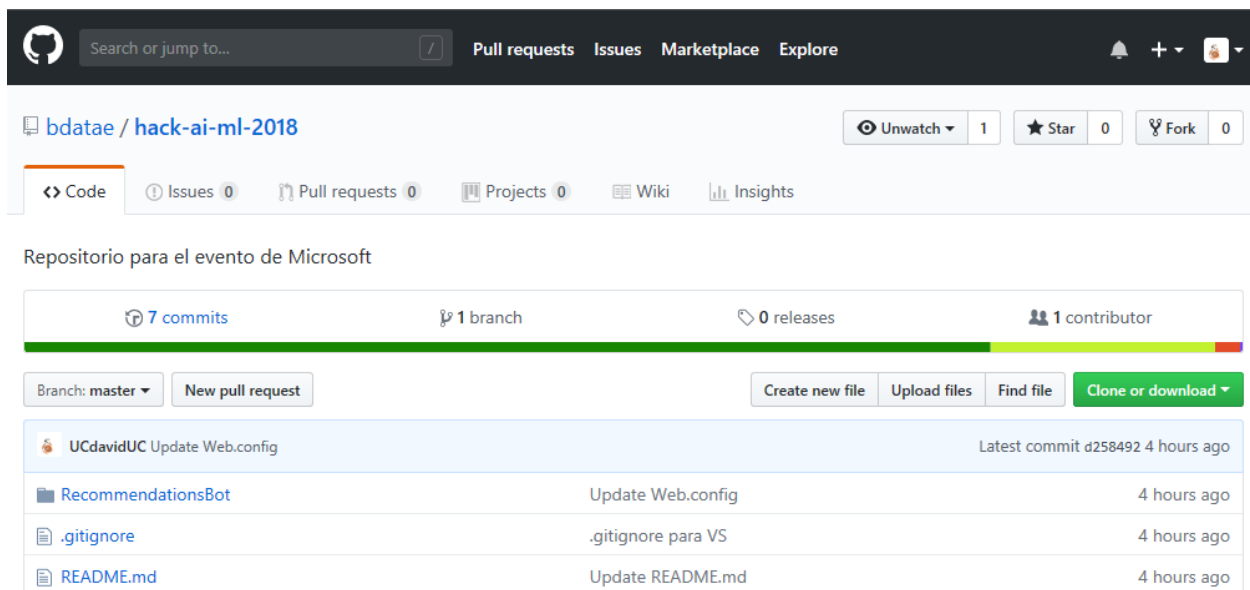
13. La captura de pantalla muestra el chat de pruebas, este chat facilita el probar los cambios realizados en Bot desde el IDE Web una vez compilada y ejecutada la aplicación.



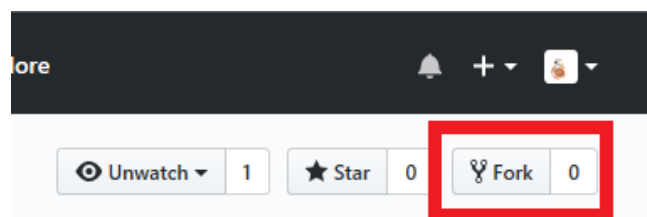
Ejercicio 3: Configuración del Bot

En la etapa de configuración conectaremos el código del Bot localizado en un repositorio de Github.com para crear un modelo de despliegue continuo. Posteriormente haremos el cambio de las cadenas de configuración del Bot, almacenamiento y servicios conectados (Bot Service, Azure Machine Learning, Storage Account y Bing Search).

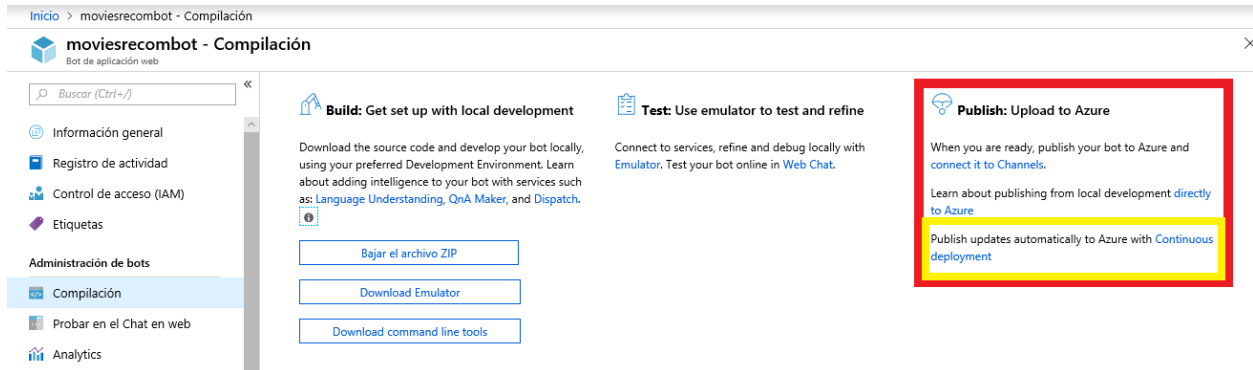
1. Crear una cuenta de Github.com o hacer login con una cuenta existente.
2. Acceder a este [link](#), el cual apunta a un repositorio con el contenido del Workshop.



3. Dentro del repositorio de Github, seleccionar la opción "Fork", esto creará un clon del espacio de desarrollo del Bot en tu cuenta de Github. Cada vez que se hagan cambios al repositorio original podrás descargarlos a tu copia del repositorio conservando los cambios que hayas realizado.



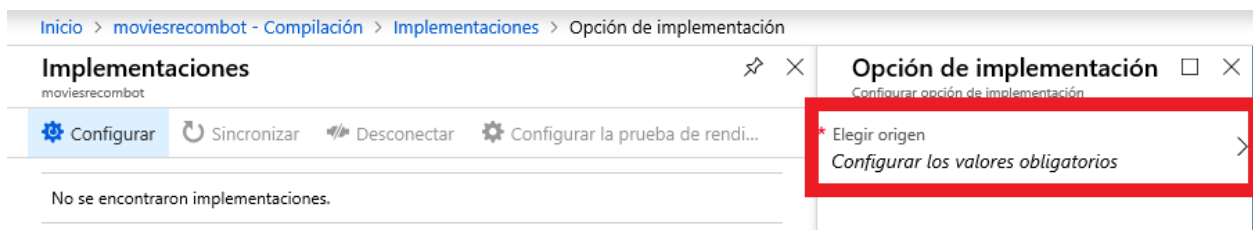
4. En el Portal de Azure, ir al recurso de "Bot de aplicación web", en la barra de herramientas de la izquierda, en el apartado de "Administración de Bots", seleccionar "Compilación". En la sección de "Publish" revisada en la sección de Revisión del Web IDE, seleccionar el link correspondiente a "Continuous deployment".



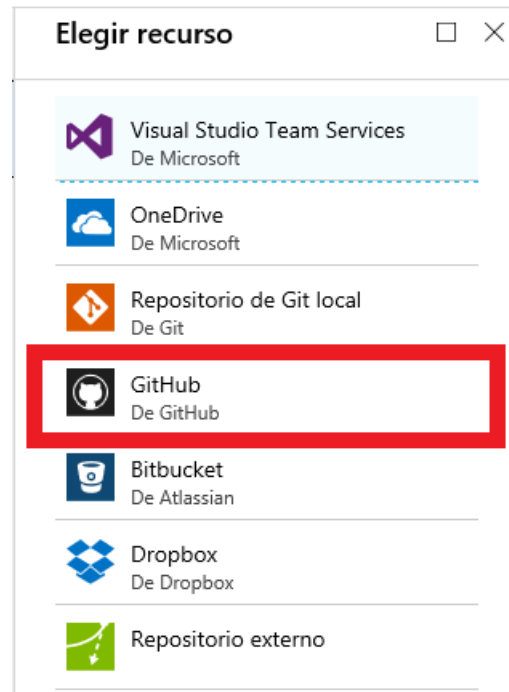
5. Al seleccionar esta opción aparecerá una pantalla con escenarios de DevOps, en caso de tener alguno, o vacía. Crearemos una cuenta de DevOps en Azure que nos permitirá conectar fuentes de código para despliegue continuo, en este caso estaremos utilizando Github. Para crear un nuevo espacio, hacer clic en "Configurar" en la esquina superior izquierda.



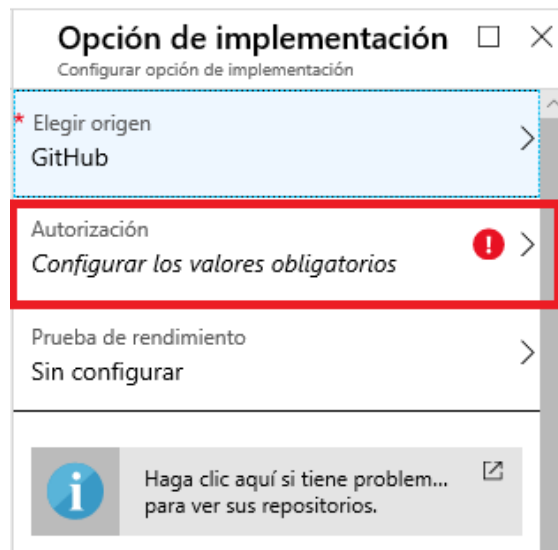
6. Se desplegará una ventana a la derecha, seleccionar "Configurar los valores obligatorios".



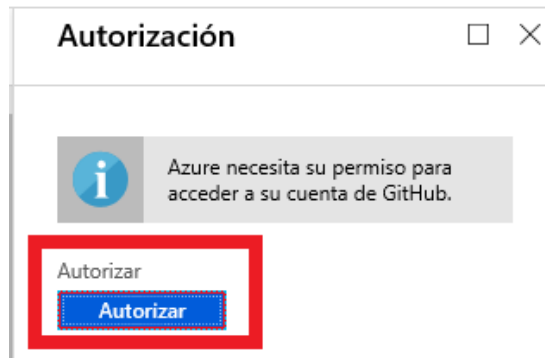
7. En la ventana "Elegir recurso", seleccionar "Github".



8. Una vez seleccionada la opción, hacer clic en la configuración de "Autorización".



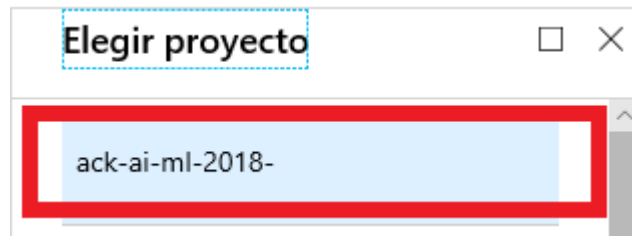
9. La autorización abre una nueva venta, en esta ventana debes completar los campos con tu nombre de usuario y contraseña de tu cuenta de Github.



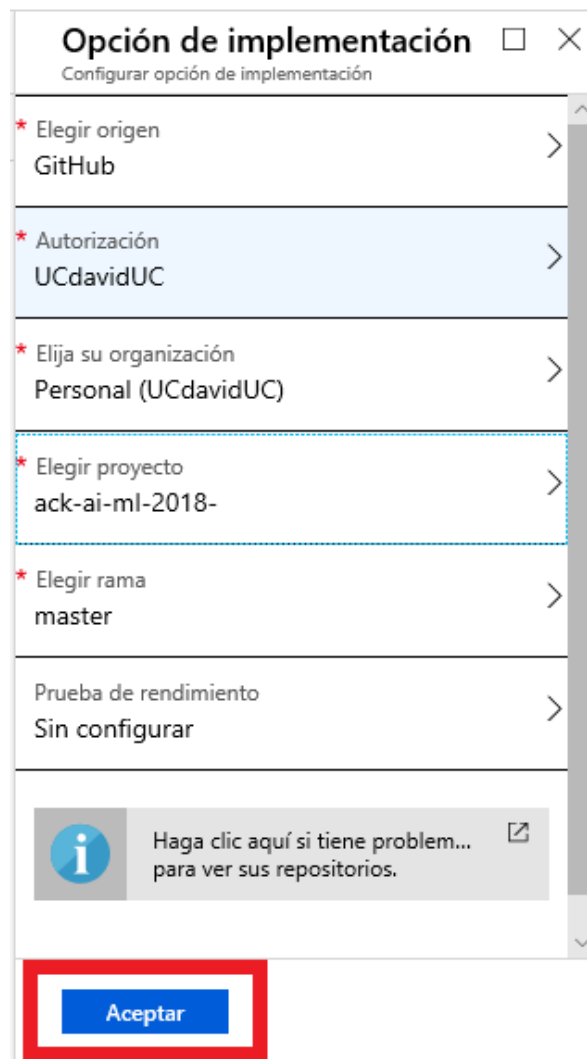
10. Una vez que son completados los campos de la forma de Github de forma correcta, es posible regresar a una vista de confirmación, hacer clic en "Aceptar".



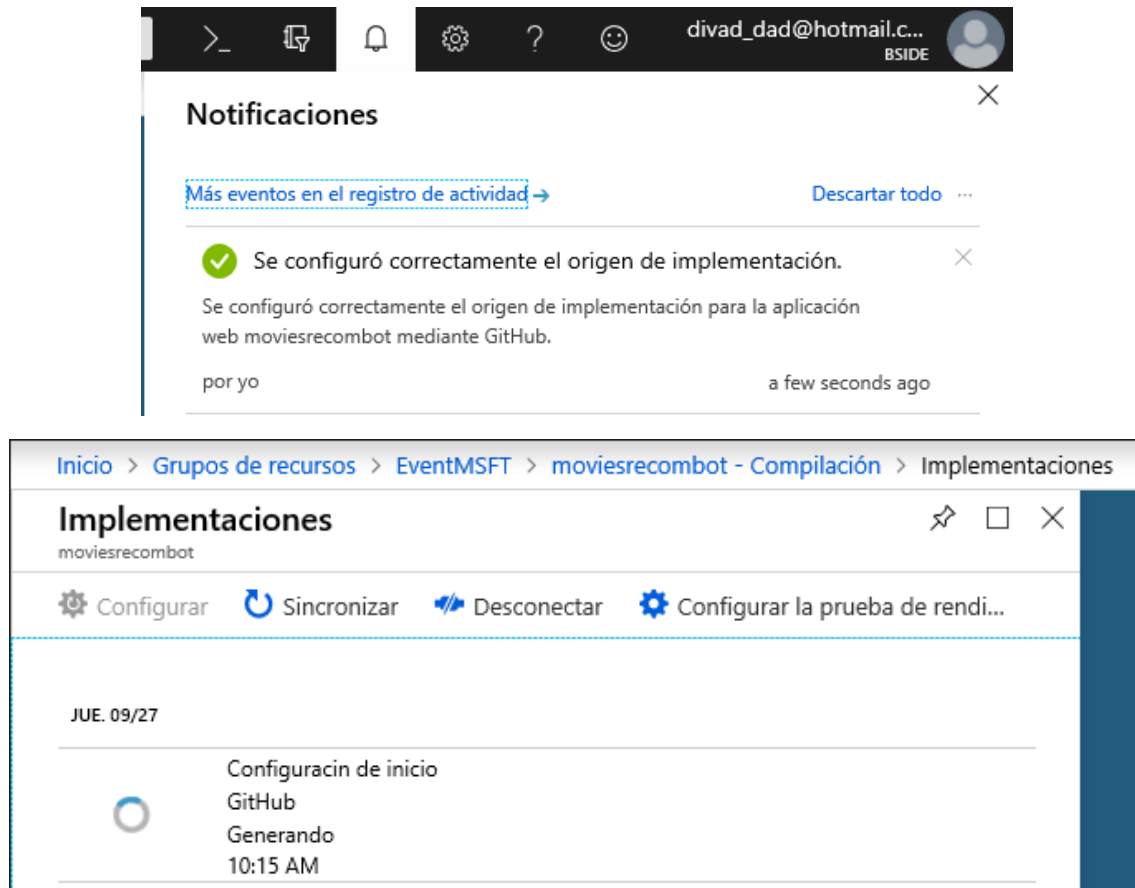
11. Una vez seleccionado, desplegará un panel con una lista de repositorios vinculados a tu cuenta de Github. Para realizar la conexión con Azure selecciona el repositorio a partir del cual se tomará el código.



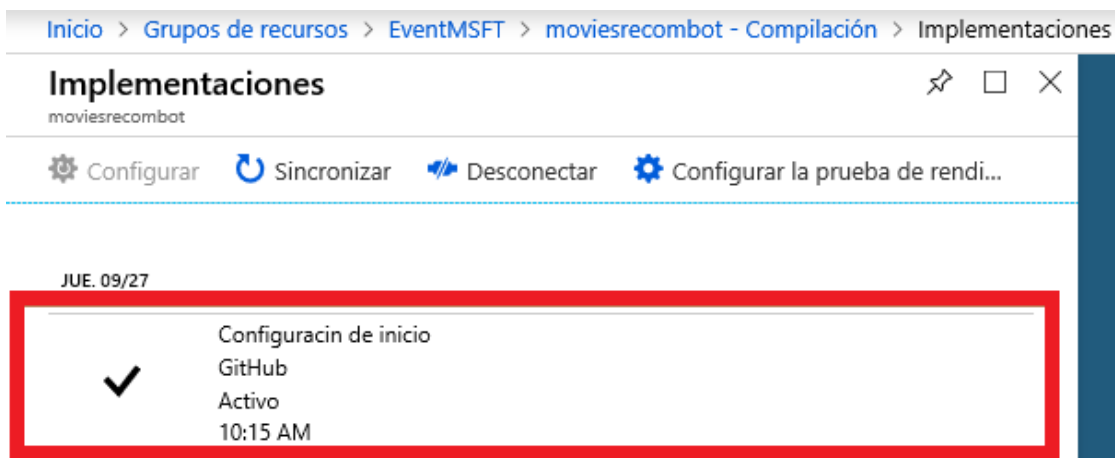
12. Una vez seleccionado, Azure toma la información de las ramas y configuración del repositorio. En este caso seleccionaremos la rama "master" y dejaremos el campo



13. Una vez seleccionado y aceptada la configuración de Github, se desplegarán notificaciones respecto al proceso de actualización contra el repositorio:



14. La confirmación se muestra con una paloma, lo cual indica que la conexión fue exitosa y que los cambios realizados en el repositorio serán tomados por Azure y se verán reflejados en actualizaciones sobre el Bot.

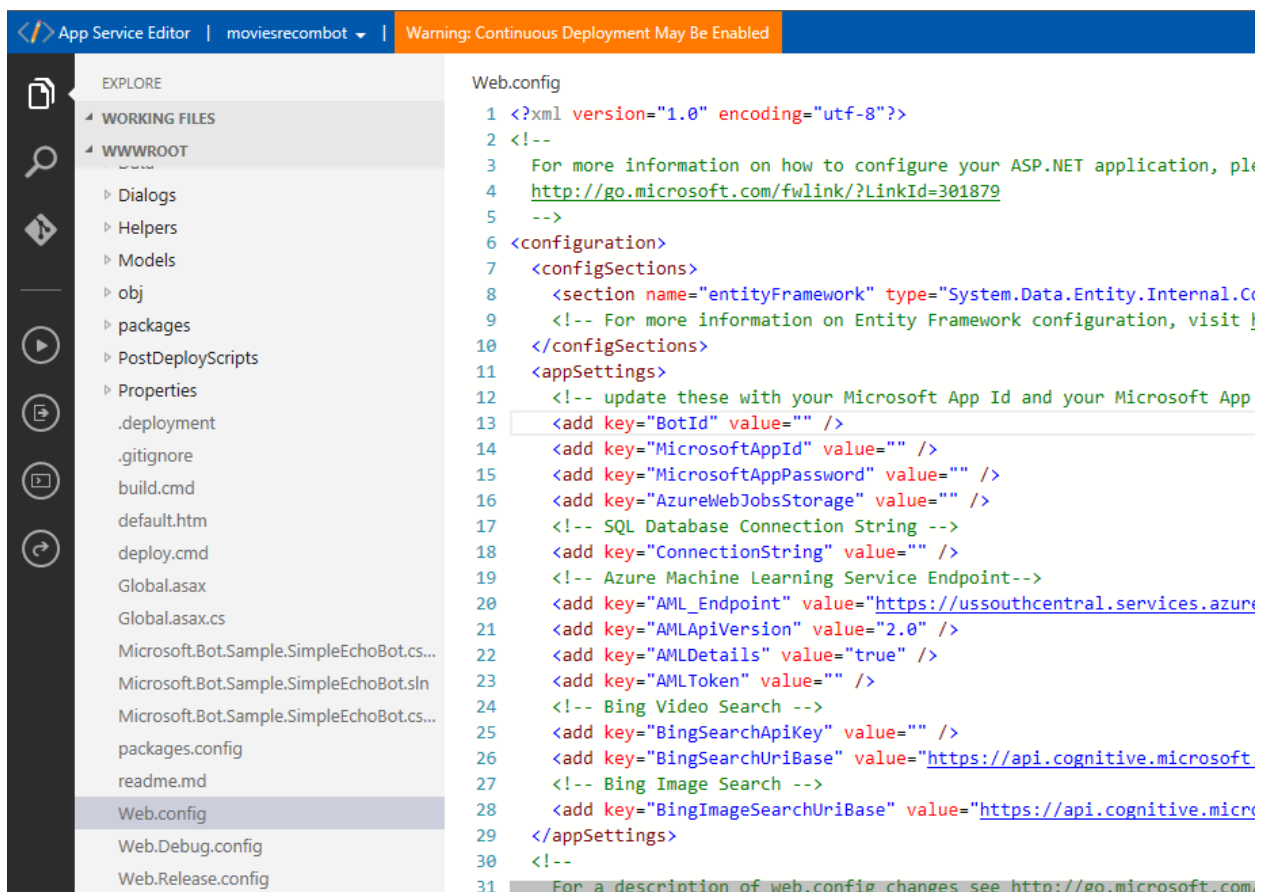


15. ¡Ya podemos probar el Bot! ... Paciencia saltamontes, necesitamos darle el origen de datos al Bot para que permita las conexiones con Azure Machine Learning y Servicios Cognitivos.
16. Para realizar la configuración, debemos entender primero de donde viene la configuración. En la barra superior se encuentra un mensaje de alerta, este mensaje indica que una vez conectado con el repositorio, Azure estará validando cambios sobre el repositorio, así que cualquier cambio realizado sobre el IDE embebido será sobrescrito por el contenido del repositorio.

ALERTA: Podríamos configurar esto en Github, sin embargo, el repositorio es público, así que eso haría que las claves lo fuesen también. ¿Cómo podemos darle la vuelta al problema?

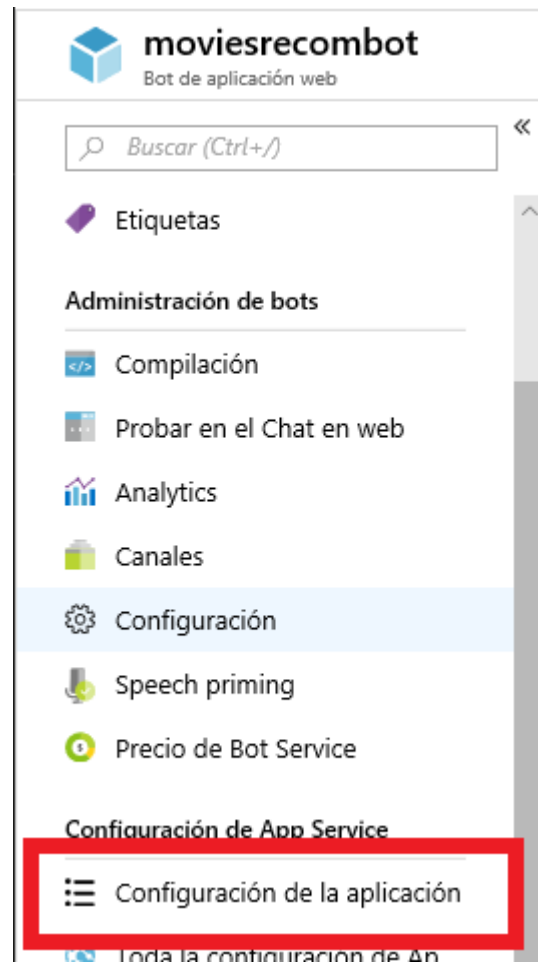
El servicio de Bot tiene una página de configuraciones que sobre escribe sobre la configuración de la aplicación de Bot permitiendo así, por una parte, tener resguardada la información en un sitio seguro dentro de Azure y por el otro tener un acceso rápido para realizar cambios.

En la imagen observamos el archivo Web.config cargado de Github sin llaves.





```
Web.config
1 <?xml version="1.0" encoding="utf-8"?>
2 <!--
3   For more information on how to configure your ASP.NET application, please visit
4   http://go.microsoft.com/fwlink/?LinkId=301879
5   -->
6 <configuration>
7   <configSections>
8     <section name="entityFramework" type="System.Data.Entity.Internal.C
9     <!-- For more information on Entity Framework configuration, visit !
10  </configSections>
11  <appSettings>
12    <!-- update these with your Microsoft App Id and your Microsoft App
13    <add key="BotId" value="" />
14    <add key="MicrosoftAppId" value="" />
15    <add key="MicrosoftAppPassword" value="" />
16    <add key="AzureWebJobsStorage" value="" />
17    <!-- SQL Database Connection String -->
18    <add key="ConnectionString" value="" />
19    <!-- Azure Machine Learning Service Endpoint-->
20    <add key="AML_Endpoint" value="https://ussouthcentral.services.azure
21    <add key="AMLApiVersion" value="2.0" />
22    <add key="AMLDetails" value="true" />
23    <add key="AMLTOKEN" value="" />
24    <!-- Bing Video Search -->
25    <add key="BingSearchApiKey" value="" />
26    <add key="BingSearchUriBase" value="https://api.cognitive.microsoft
27    <!-- Bing Image Search -->
28    <add key="BingImageSearchUriBase" value="https://api.cognitive.micr
29  </appSettings>
30  <!--
31  For a description of web.config changes see http://go.microsoft.com
```

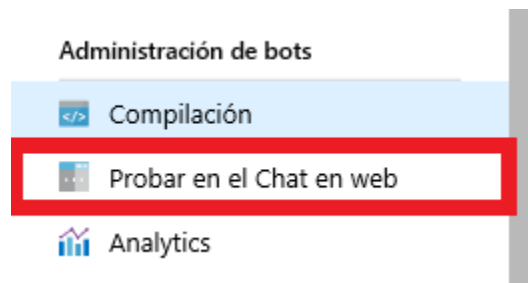
17. Para realizar la configuración del servicio vamos a la ventana de configuraciones, en el segmento de "Configuración de App Service", seleccionamos "Configuración de la aplicación".



18. Dentro del archivo de configuración establecemos las claves de los servicios conectados. Hacemos clic en el botón "Guardar", Azure te notificará sobre la actualización, la cual ocurre en cuestión de un par de segundos.

 Guardar  Descartar			
AML_Endpoint	<input type="checkbox"/>		×
AMLApiVersion	<input type="checkbox"/>		×
AMLDetails	<input type="checkbox"/>		×
AMLTOKEN	<input type="checkbox"/>		×
AzureWebJobsDashboard	<input type="checkbox"/>		×
AzureWebJobsStorage	<input type="checkbox"/>		×
BingImageSearchUriBase	<input type="checkbox"/>		×
BingSearchApiKey	<input type="checkbox"/>		×
BingSearchUriBase	<input type="checkbox"/>		×
BotEnv	<input type="checkbox"/>		×
BotId	<input type="checkbox"/>		×
BotOpenIdMetadata	<input type="checkbox"/>		×
BotStateEndpoint	<input type="checkbox"/>		×
ConnectionString	<input type="checkbox"/>		×
MicrosoftAppId	<input type="checkbox"/>		×

19. Con el código corriendo y la configuración realizada, es momento de probar el servicio.
En la sección de configuración del servicio de Bot, seleccionamos nuevamente "Probar en el Chat en web".



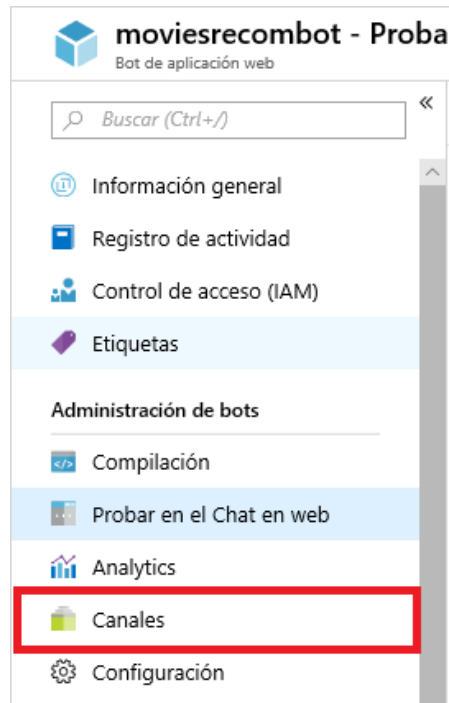
20. Escribamos "Hola" al Bot y probemos la respuesta.



21. ¡Felicidades! Ya tienes tu primer Bot inteligente.



Ejercicio 4: Publicar el Bot en una página web

1. Dentro de la configuración del servicio de Bot, seleccionar "Canales".







2. Dentro de la ventana de "Canales", seleccionar el link de "Editar" para el "Web Chat".

Conectarse a canales

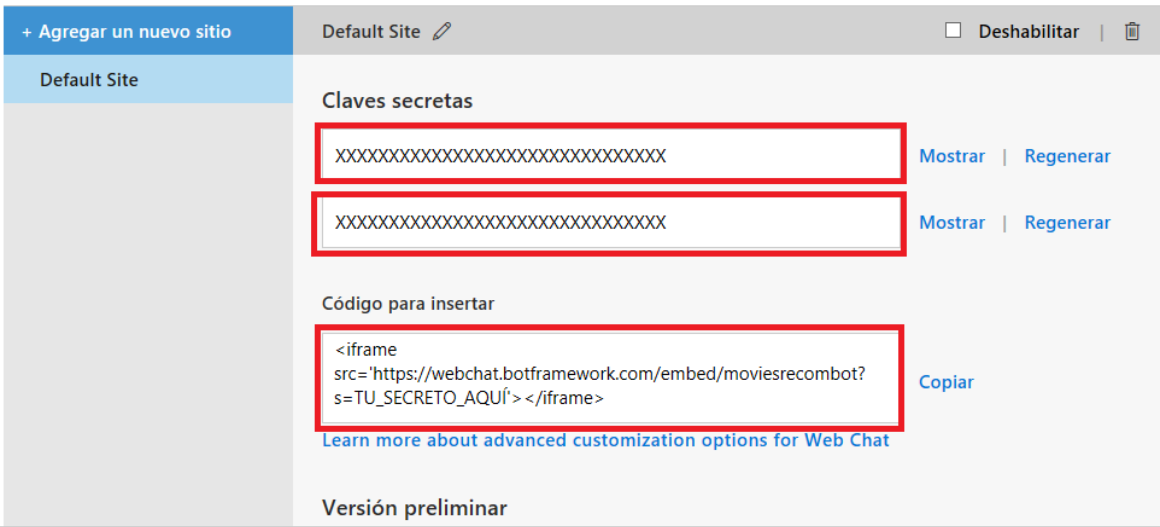
Nombre	Estado	Publicado	
 Web Chat	En funcionamiento	--	Editar 

[Obtener los códigos para insertar de los bots](#)

Agregar un canal destacado

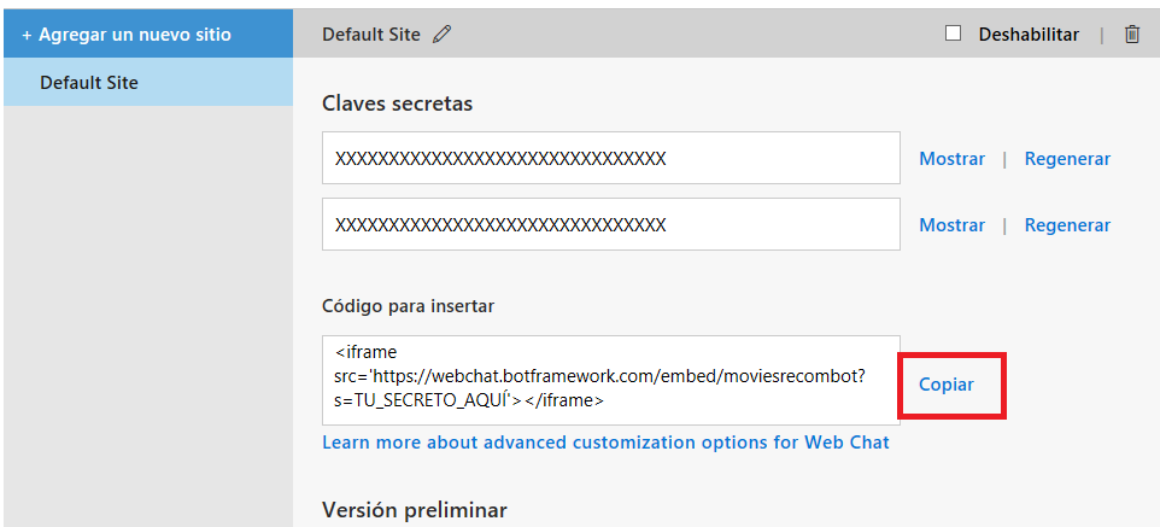


3. En la página de "Configurar Web Chat" se muestran "Claves Secretas" y "Código para Insertar". Este último es el iframe embebido en el archivo .htm de inicio.



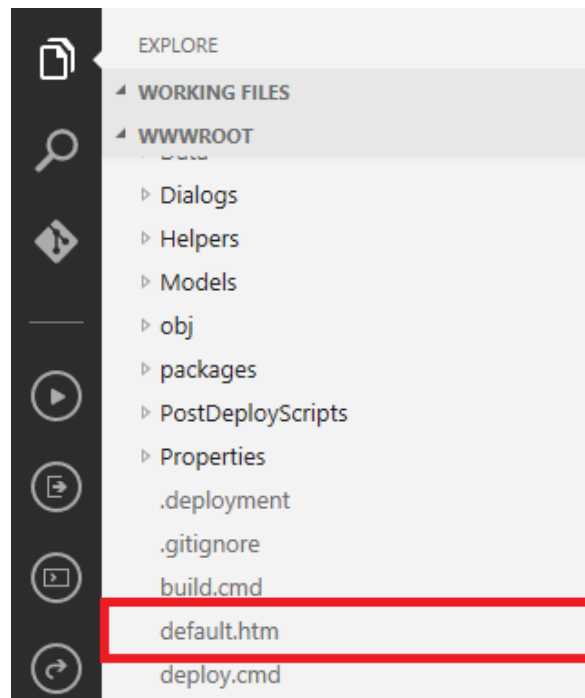
The screenshot shows the 'Configurar Web Chat' interface. On the left, there's a sidebar with a blue button '+ Agregar un nuevo sitio' and a 'Default Site' section. The main area is titled 'Default Site' and contains two sections: 'Claves secretas' and 'Código para insertar'. Under 'Claves secretas', there are two input fields, each containing a long string of 'X's. To the right of each field are links 'Mostrar' and 'Regenerar'. The 'Código para insertar' section contains an iframe code snippet: `<iframe src='https://webchat.botframework.com/embed/moviesrecombot?s=TU_SECRETO_AQUÍ'></iframe>`. To the right of the code is a 'Copiar' link. Below the code is a link 'Learn more about advanced customization options for Web Chat'. At the bottom of the main area is the text 'Versión preliminar'. A blue button 'Listo' is located at the bottom left of the interface.

4. Hacemos clic en "Copiar". Para practicidad, abrimos una nueva Tab en la que abriremos otra instancia del Portal de Azure.



This screenshot is identical to the previous one, showing the 'Configurar Web Chat' interface. The 'Copiar' link next to the embed code is highlighted with a red rectangle, indicating the next step in the process.

5. En el nuevo Tab navegamos a nuestro servicio de Bot. Dentro de las opciones del servicio seleccionamos "Compilación" -> "Abrir el editor de código en línea". Una vez desplegado el código, buscamos dentro del directorio raíz mostrado el archivo llamado "default.htm".



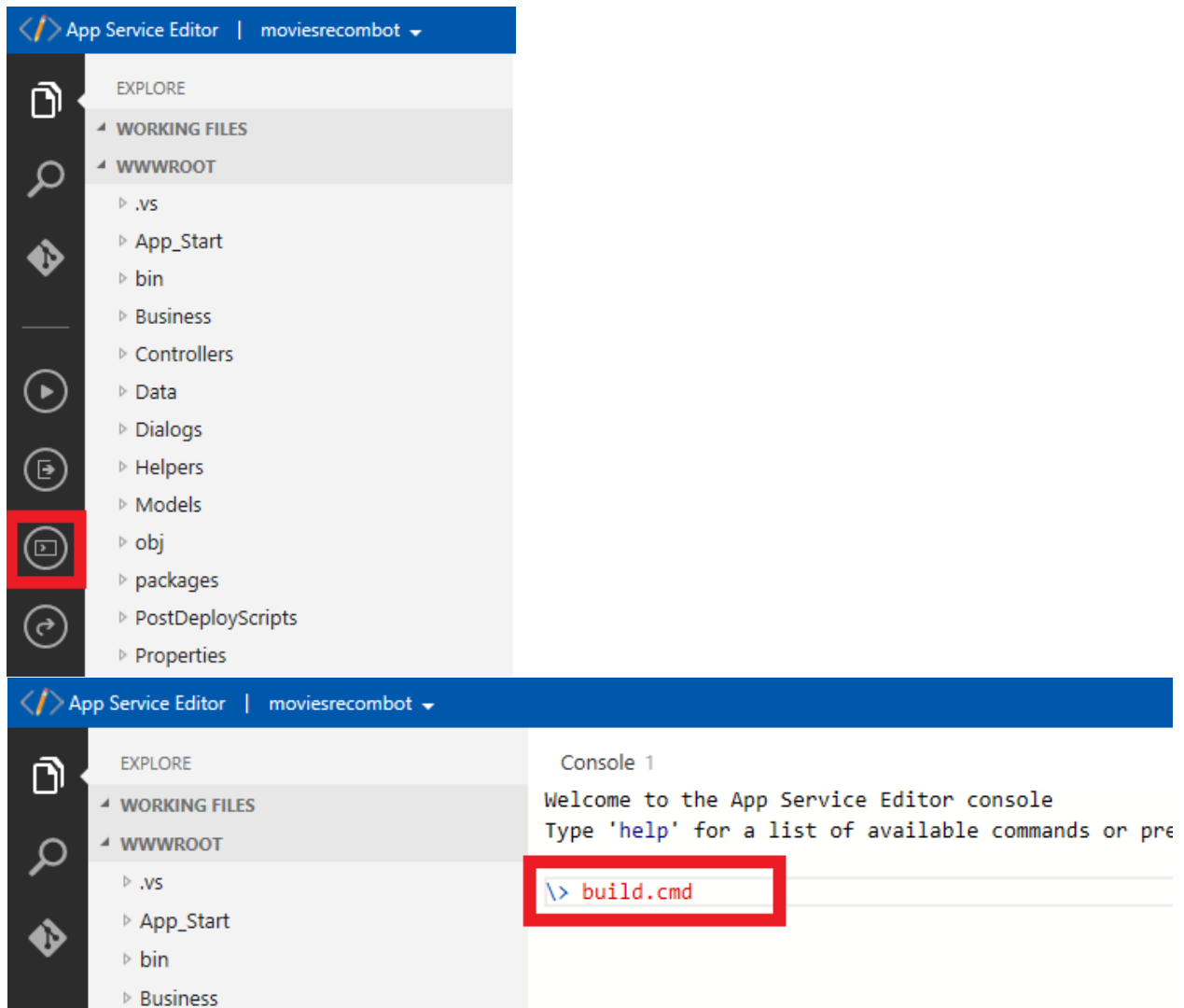
6. Dentro del código del archivo embeberemos nuestro tag de iframe con el valor correspondiente.

The image shows the App Service Editor interface with the 'default.htm' file open. The code is as follows:

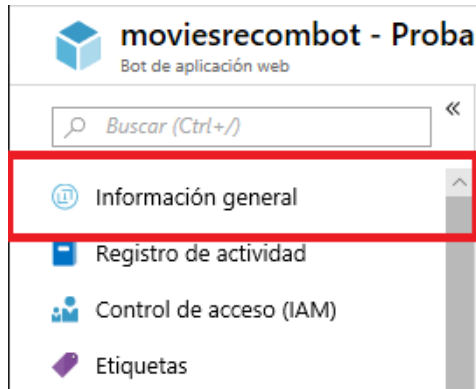
```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Las maravillas del cine</title>
5   <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css"
6   <meta charset="utf-8" />
7 </head>
8 <body style="font-family:'Segoe UI';">
9   <div class="container">
10    <h1 style="font-weight:400;">Las maravillas del cine</h1>
11    <p>La mejor recomendación para tí.</p>
12    <table class="table">
13      <tbody>
14        <tr>
15          <td>
16            <p style="text-align: right;">
17              <iframe src='https://webchat.botframework.com/embed/moviesrecombot?s=r6J3w7'
18            </p>
19          </td>
20        </tr>
21      </tbody>
22    </table>
23  </div>
24 </body>
25 </html>
26
```

The iframe tag on line 17 is highlighted with a red rectangular box.

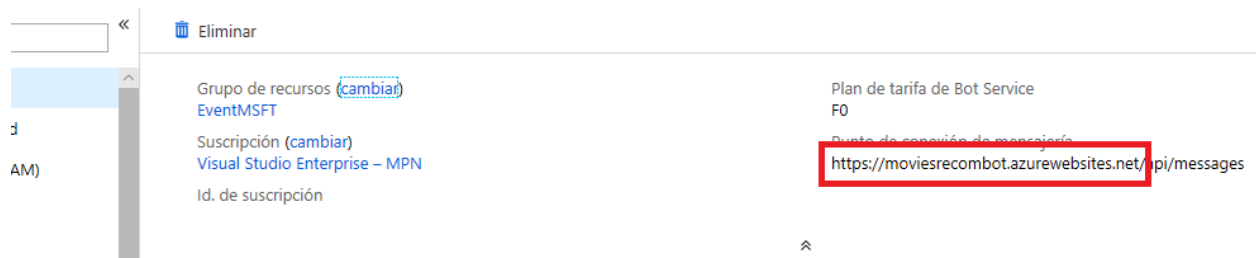
7. Citando el paso de compilación, debemos compilar y correr el código del Bot cuando realicemos el cambio.



8. Regresamos al Tab de "Canales". De la barra de configuración de la izquierda, seleccionamos la opción de "Información general".



9. Dentro de la ventana copiamos el "Punto de conexión de mensajería". De la Url completa borramos "api/messages". Para probar, pegamos la Url en el navegador.



10. El Bot ahora se encuentra embebido en un sitio Web, de igual forma podemos integrar más canales de comunicación como puede ser Skype o Microsoft Teams.