Hand son lab Creación de un Bot con QnA

Mario Cortés Flores

[mariocortesflores@hotmail.com](mailto:mariocortesflores@hotmail.com)

v1

Contenido

[Objetivos del laboratorio 2](#_Toc530089935)

[Prerequisitos 2](#_Toc530089936)

[Creación de un knowledge base con QnA 2](#_Toc530089937)

[Crear un BOT que consume QnA 7](#_Toc530089938)

[Publicación del BOT desde Visual Studio 10](#_Toc530089939)

[Instalar App Studio 16](#_Toc530089940)

[Referencias 23](#_Toc530089941)

# Objetivos del laboratorio

El objetivo del laboratorio es el de proporcionar los conocimientos para crear un bot que responda a preguntas sobre Microsoft Teams a través del servicio QnA e integrado desde Microsoft Teams.

# Prerequisitos

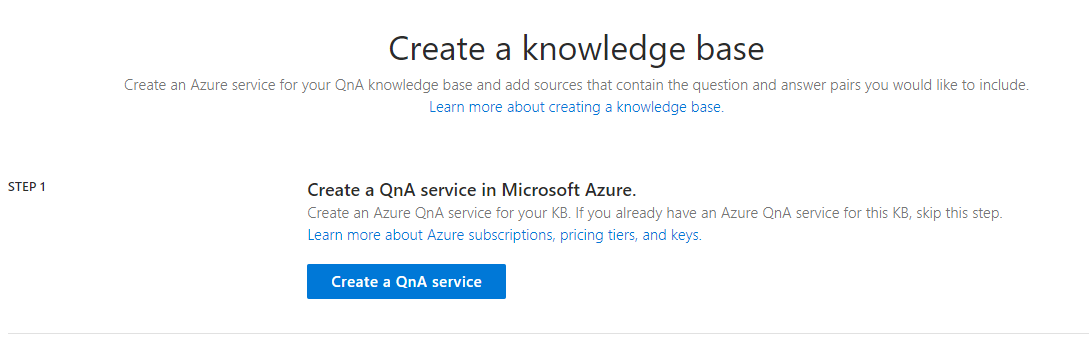
Será necesario disponer:

* Disponer de una suscripción de Office 365.
* Una subscripción de Microsoft Azure, si no se dispone de una suscripción, puede adquirir una trial desde <https://ms.portal.azure.com>. Es recomendable asociar la suscripción de Azure a la de Office 365.
* Visual Studio 2017.
* Git

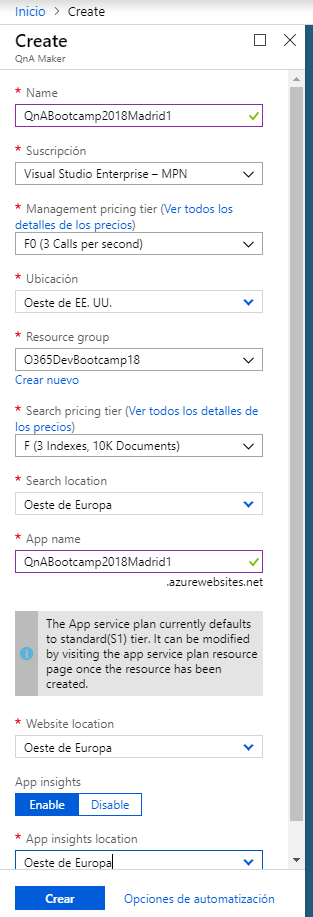
# Creación de un knowledge base con QnA

Crearemos un bot que responda a determinadas preguntas sobre el uso de Microsoft Teams.

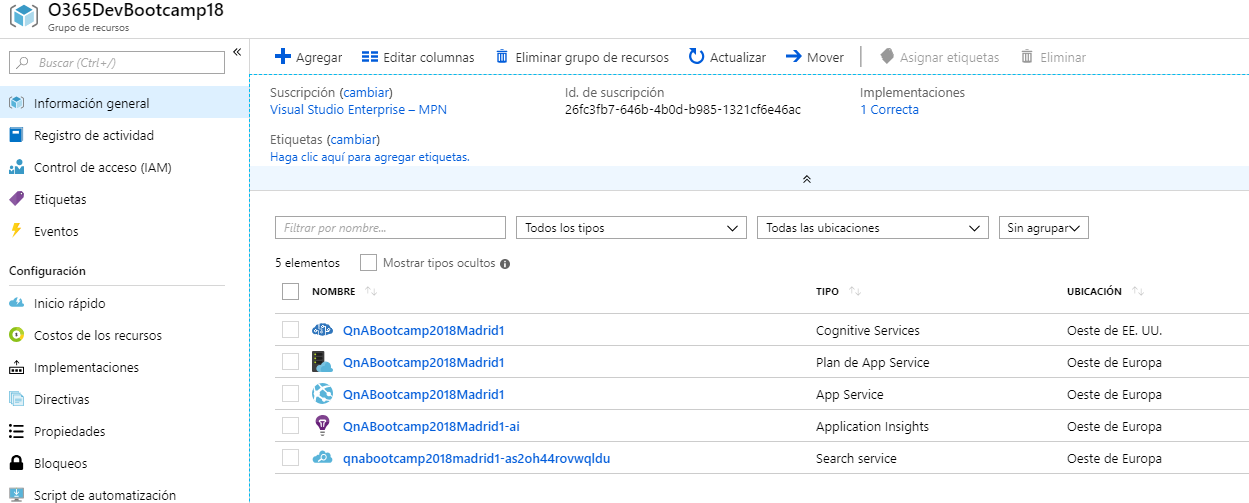
* Accedemos al portal [QnA Maker](https://www.qnamaker.ai/) e iniciamos sesión con una cuenta Microsoft y accedemos a “Create a knowledge base”. Se abrirá entonces un asistente donde tendremos que crear el servicio de QnA en la suscripción de Azure y cargar las preguntas-respuestas.
  + Primero creamos un QnA service en Microsft Azure pulsando sobre el botón “Create QnA Service”:



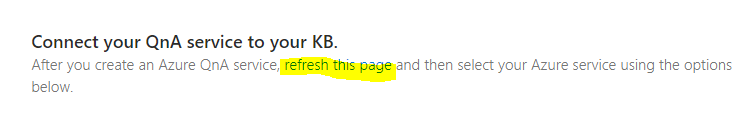
* + Se abrirá entonces una pestaña con el formulario de creación del servicio de QnA en Azure. Si no estaba logado en Azure previamente, inicie sesión con el usuario de Microsoft Azure y seleccione la suscripción en caso de tener varias disponibles.
  + Complete los datos del formulario indicando el nombre y modelo de precios, elegiremos para pruebas el plan Free o “F”. **NOTA**: Solo podremos tener un bot para un plan Free.



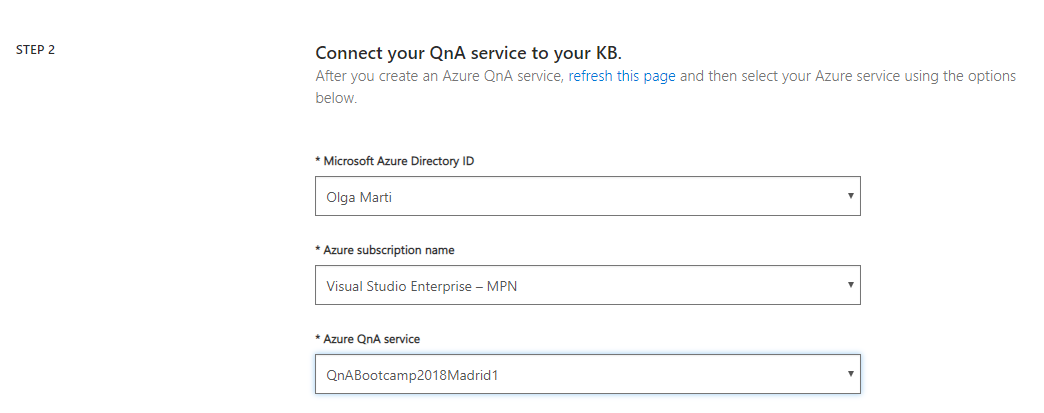
* + Una vez creado el servicio, debería aparecer algo similar a la siguiente imagen:



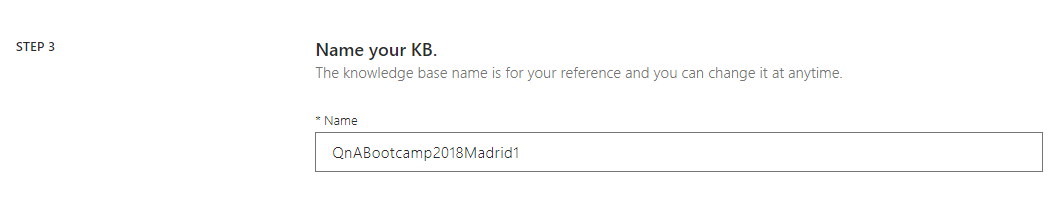
* + A continuación, vuelva a la pestaña de QnA Maker y refresque la página:



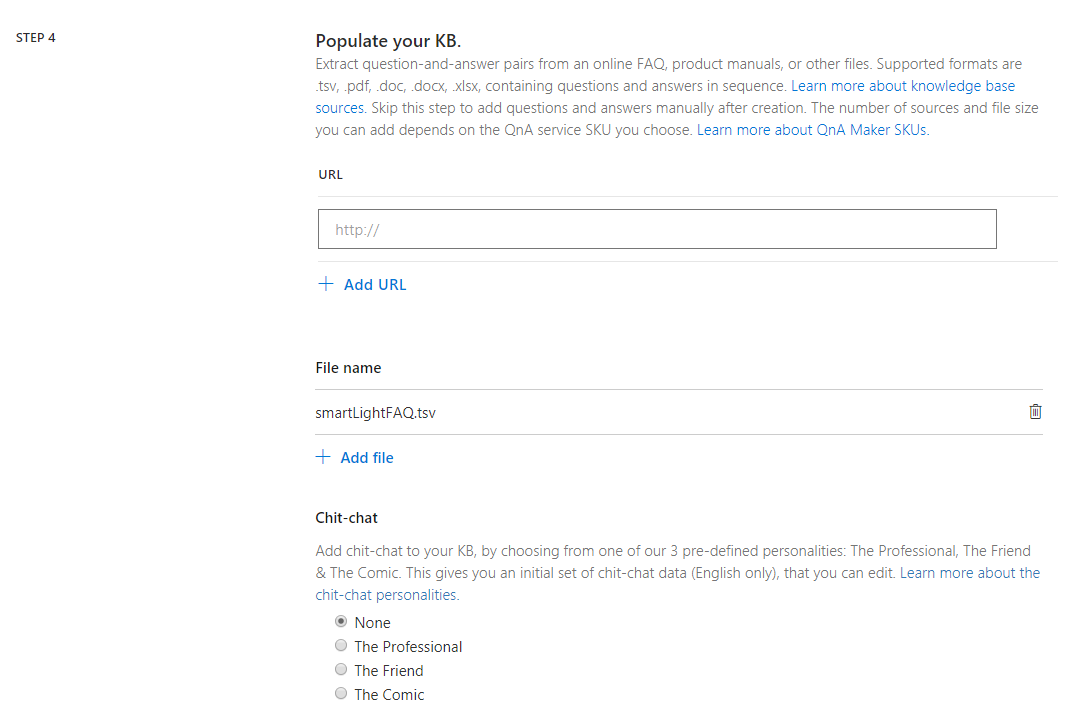
* + Deberá aparecer entonces el nuevo servicio de QnA creado:



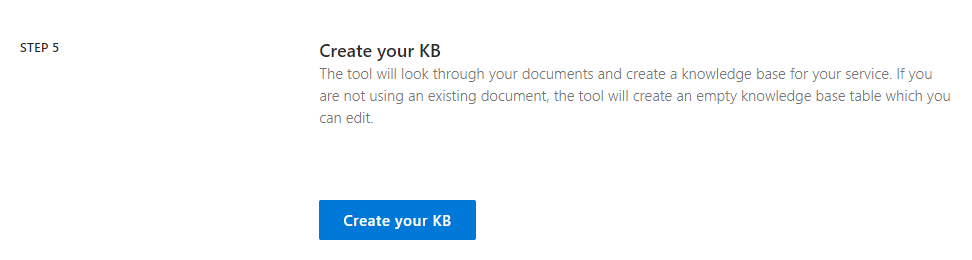
* + En el paso 3 indicamos el nombre del KB que se creará dentro del QnA Maker



* + Indicaremos el fichero de preguntas y respuestas, para el ejemplo, selecicone el fichero que se adjunta en el laboratorio smartLightFAQ.tsv:



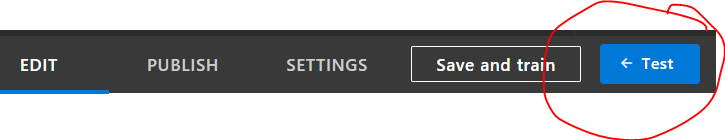
* + Pulse en “Create your KB”:



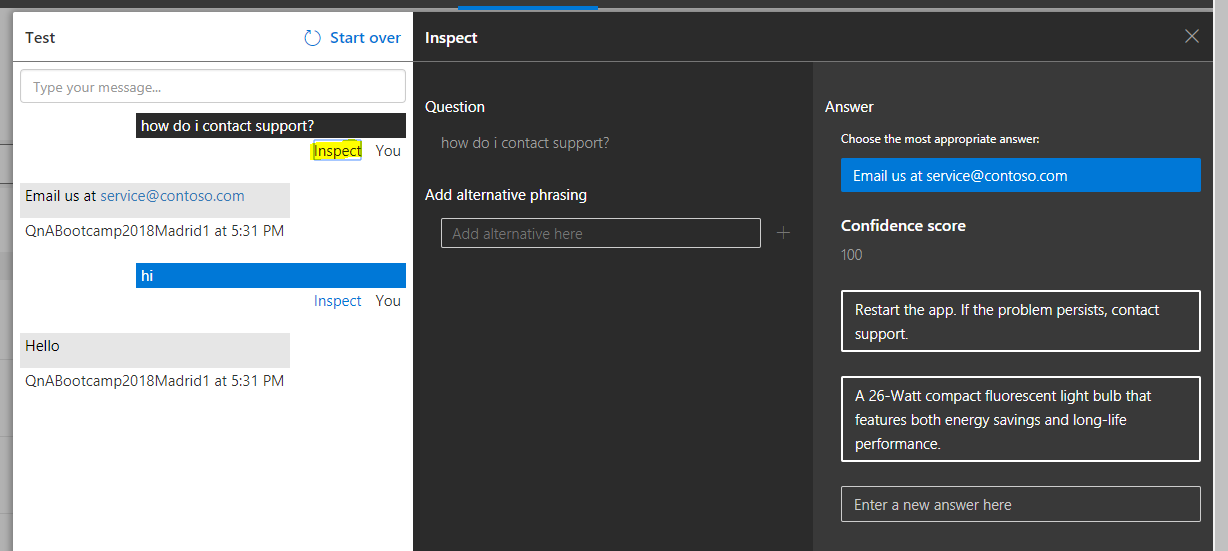
* Una vez creado, se mostrará el contenido de KB:

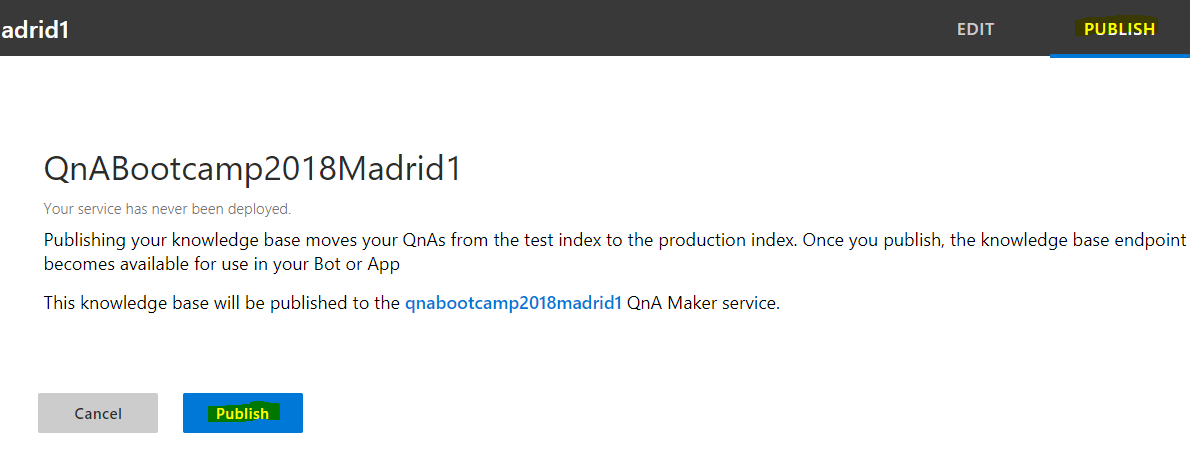


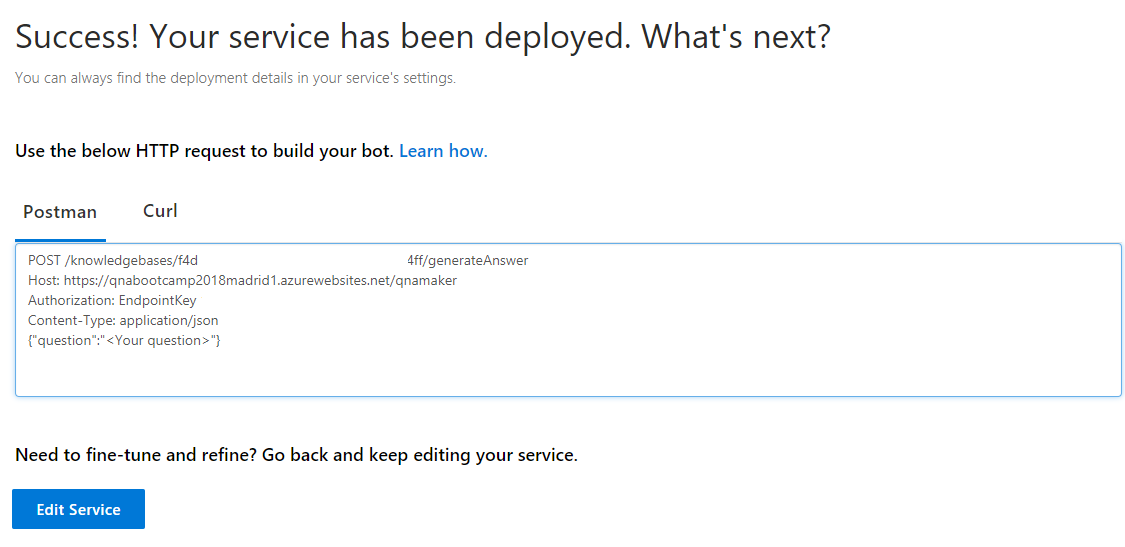
* Podrá mantener las preguntas-respuestas pulsando en “Add QnA Pair” e introduzca los nuevos valores y pulse en “Save and train” una vez finalizado.
* Para probar el bot pulse “Test”



* Aparecerá entonces un panel donde poder conversar con el bot y probar las respuestas. Podrá refinar las respuestas pulsando en “Inspect”, pudiendo indicar respuestas alternativas o seleccionar una de la lista disponible.



* Una vez finalizado el entrenamiento, habrá que publicar el bot para que pueda consumirse desde un cliente. Pulse en “Publish”:
* 
* Una vez creado, se mostrará una página con la información del servicio:



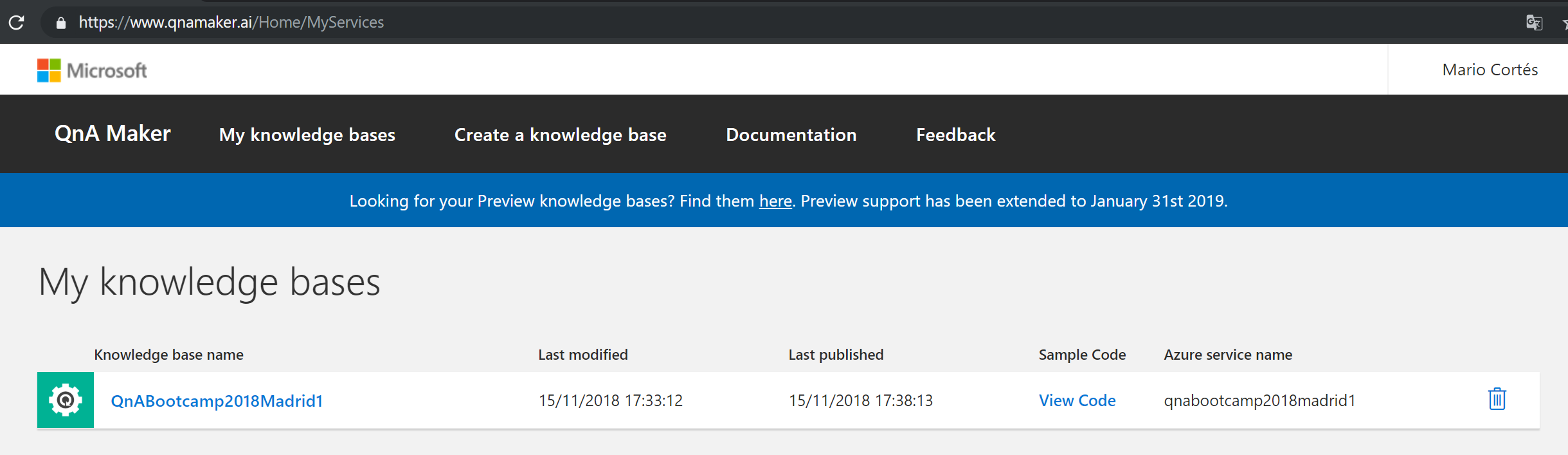
# Crear un BOT que consume QnA

Descargaremos el ejemplo de Bot de QnA desde <https://github.com/Microsoft/BotBuilder-Samples/tree/master/samples/csharp_dotnetcore/11.qnamaker>

* Clonaremos el proyecto con el siguiente comando de Git:

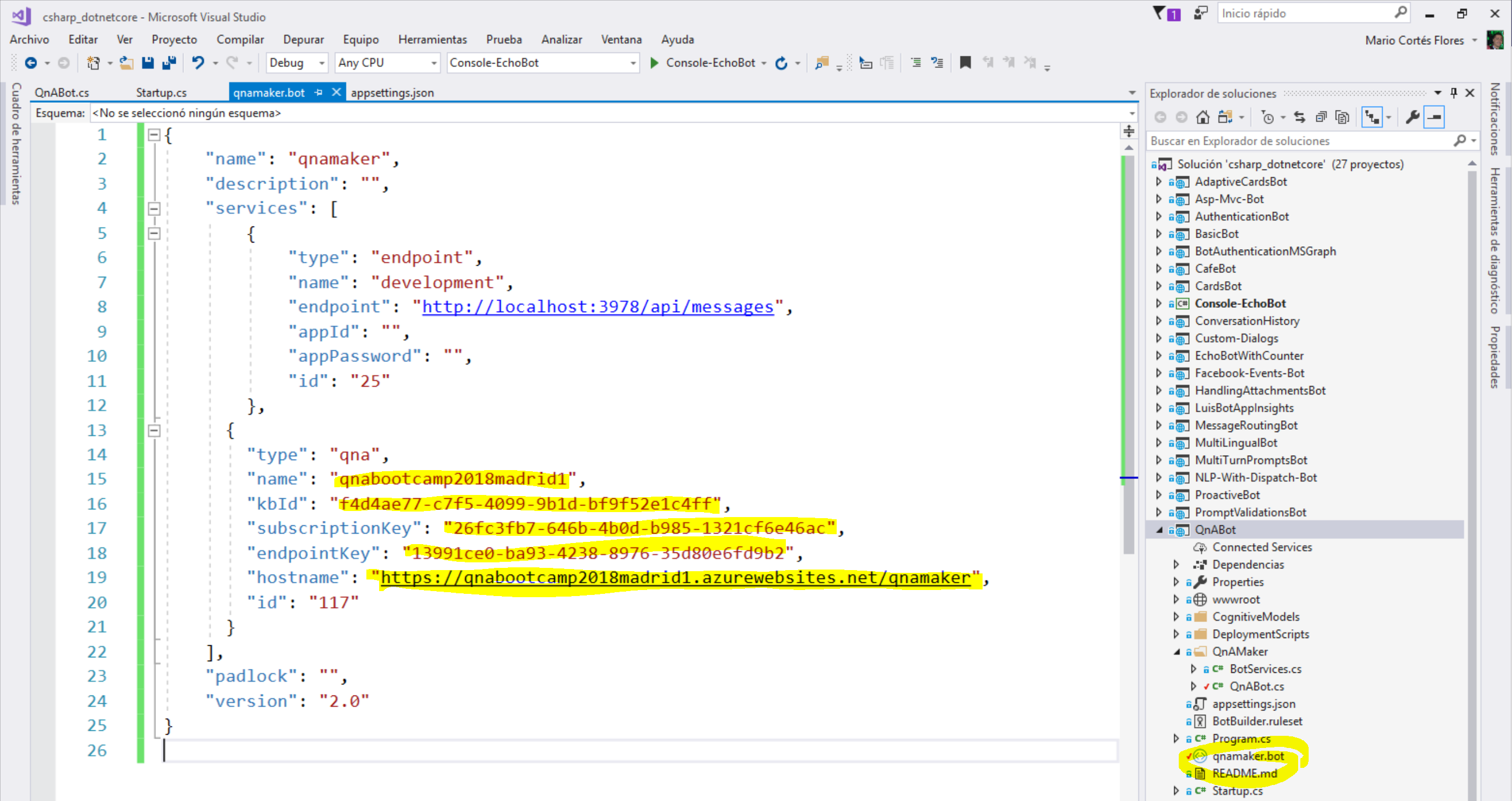
|  |
| --- |
| git clone https://github.com/Microsoft/botbuilder-samples.git |

* Una vez descargado el proyecto, abriremos la solución csharp\_dotnetcore.sln y navegaremos al proyecto “QnABot”.
* Editaremos el fichero “qnamaker.bot” con los valores **kbId** y **endpointKey** del servicio de QnA previamente creado.
  + Para obtener estos códigos, volveremos al QnA Maker y pulsaremos en “View Code” de nuestro servicio:

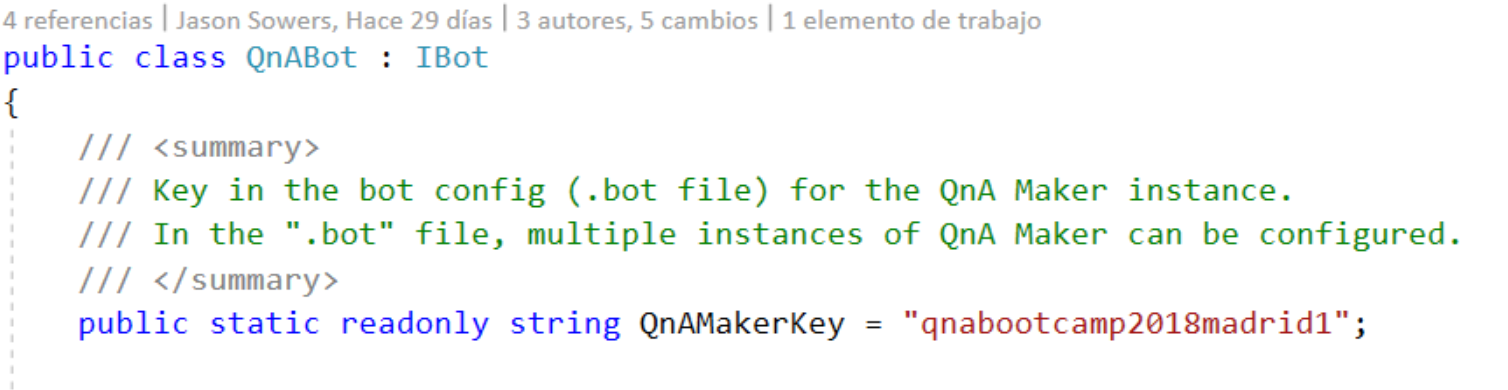


* Copiaremos los valores marcados
  + KbId
  + EndPoint Id
  + Suscription Id, será el código de suscripción de Azure
  + Name, el nombre del bot

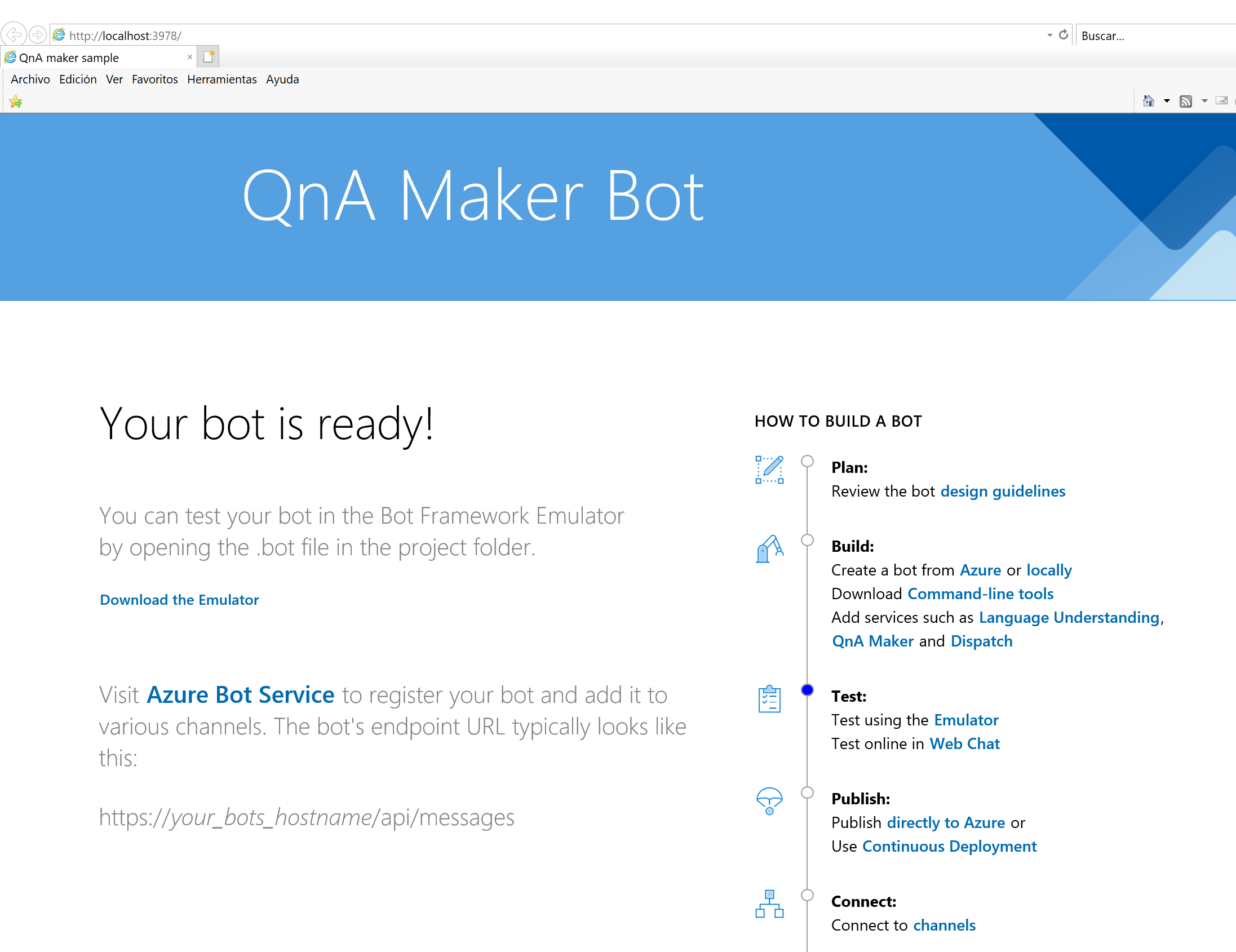




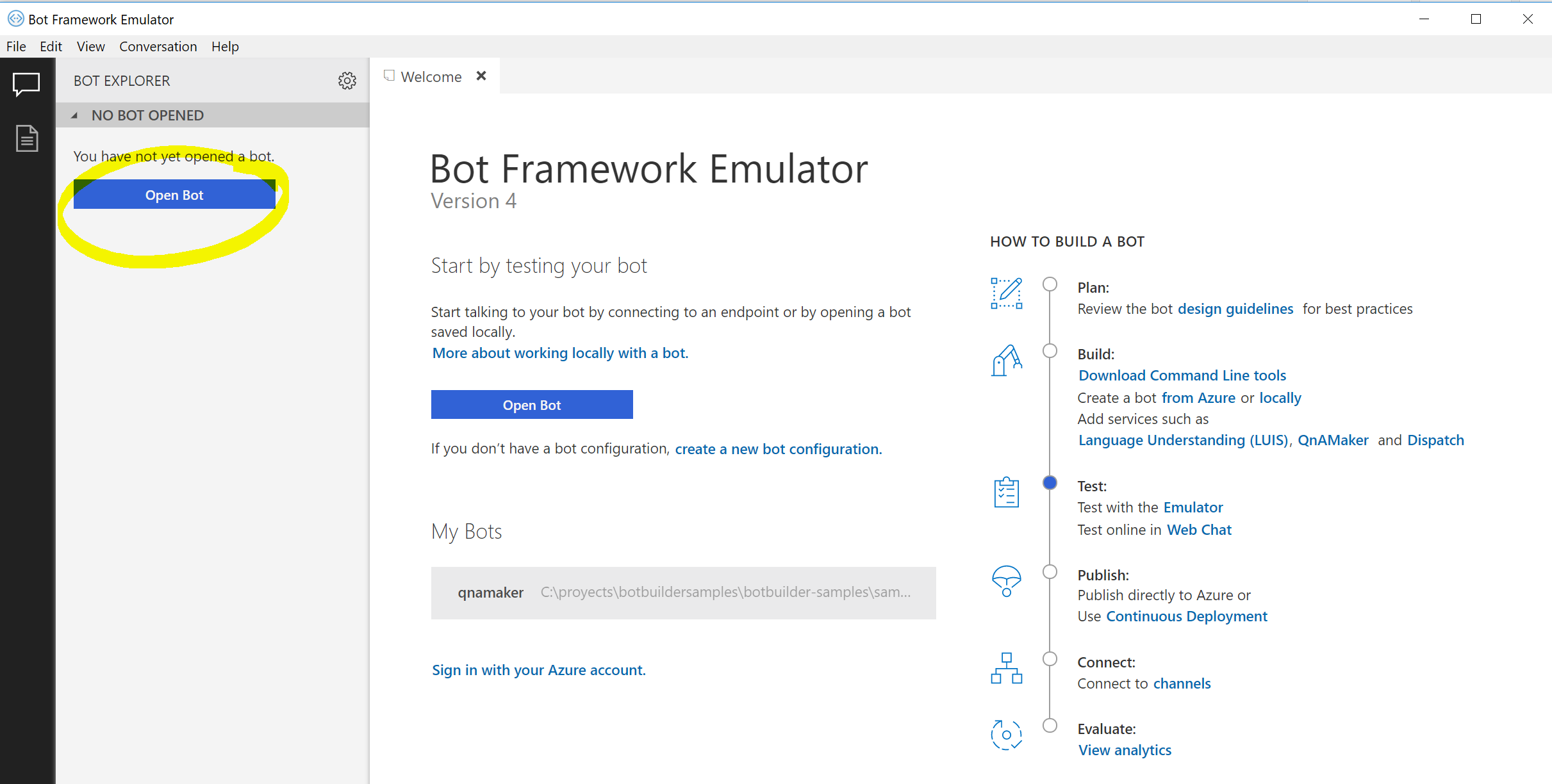
* Modificar la clase QnABot con el nombre del bot:

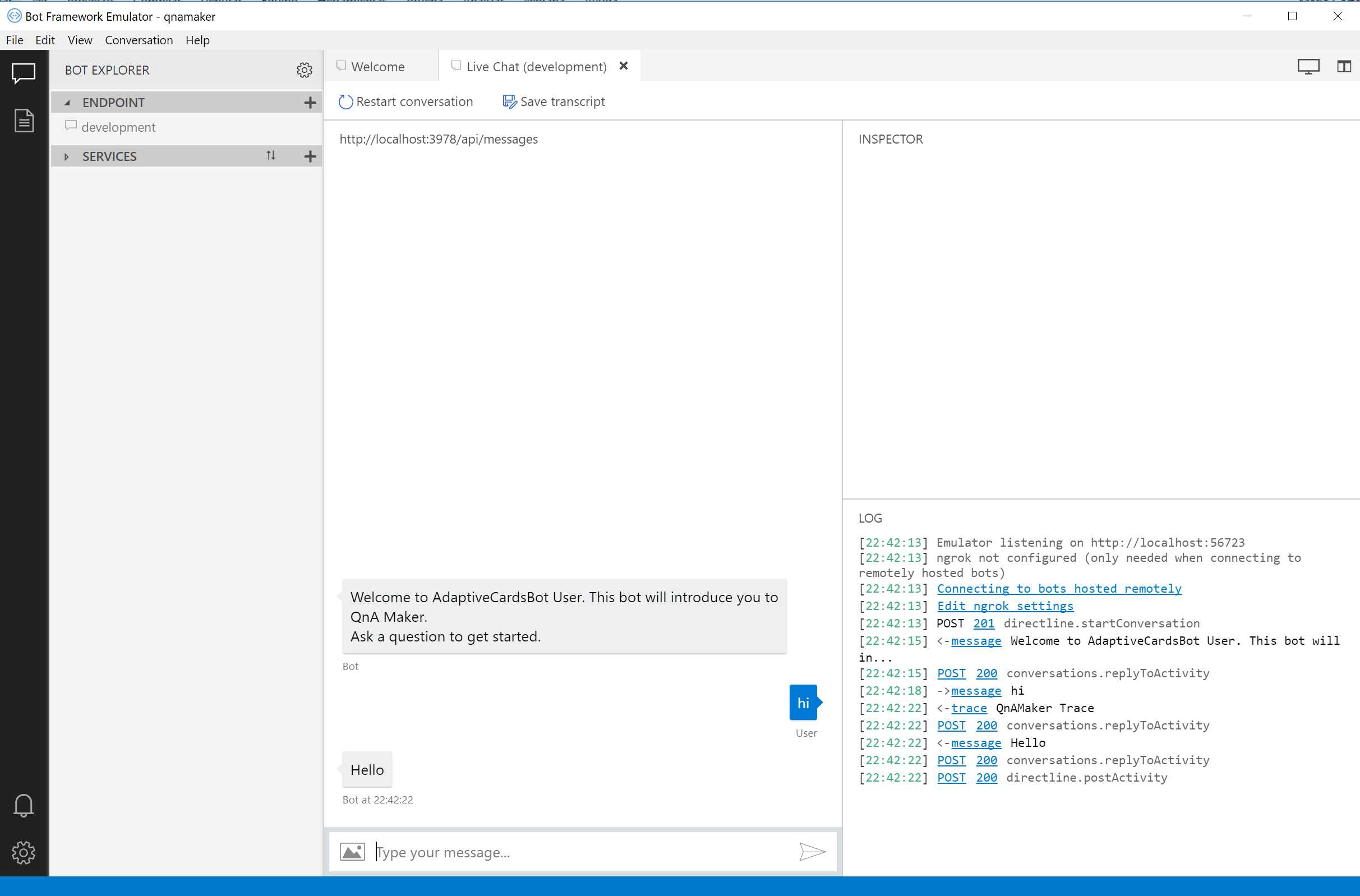


* Probaremos el proyecto compilando desde Visual Studio y ejecutando solo el proyecto (seleccionamos el proyecto->botón derecho->depurar>iniciar nueva instancia). Si se abre una ventana de navegador con el siguiente aspecto, el bot estará listo para probarse. En caso contrario, habrá que revisar los códigos introducidos.



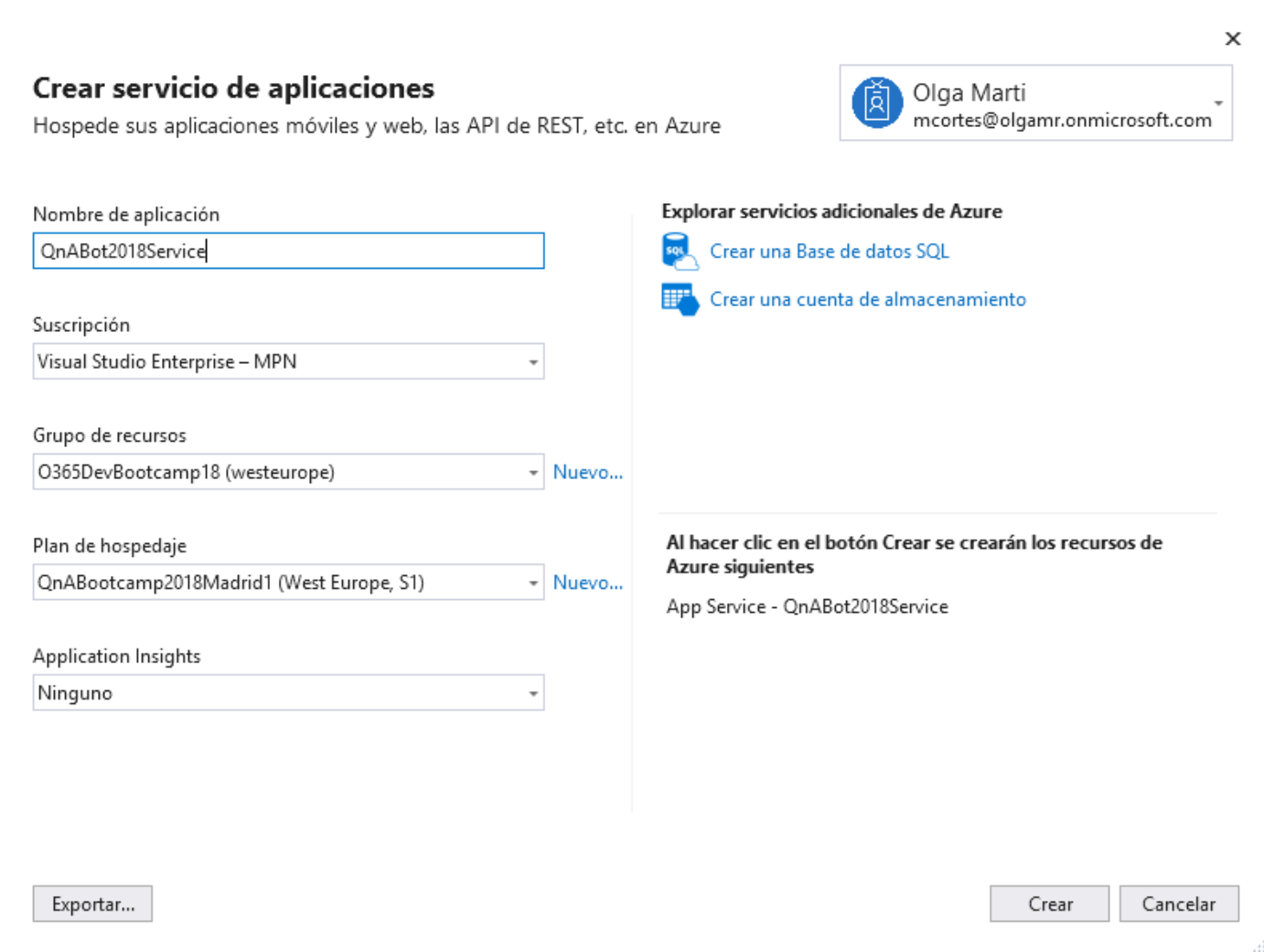
* Podemos probar nuestro bot en local utilizando el emulador. Para ello, descargue el emulador <https://github.com/microsoft/botframework-emulator>. Al iniciarlo, pulsar en “Open bot” y seleccionar el fichero qnmaker.bot del proyecto y comenzar a conversar desde el chat:



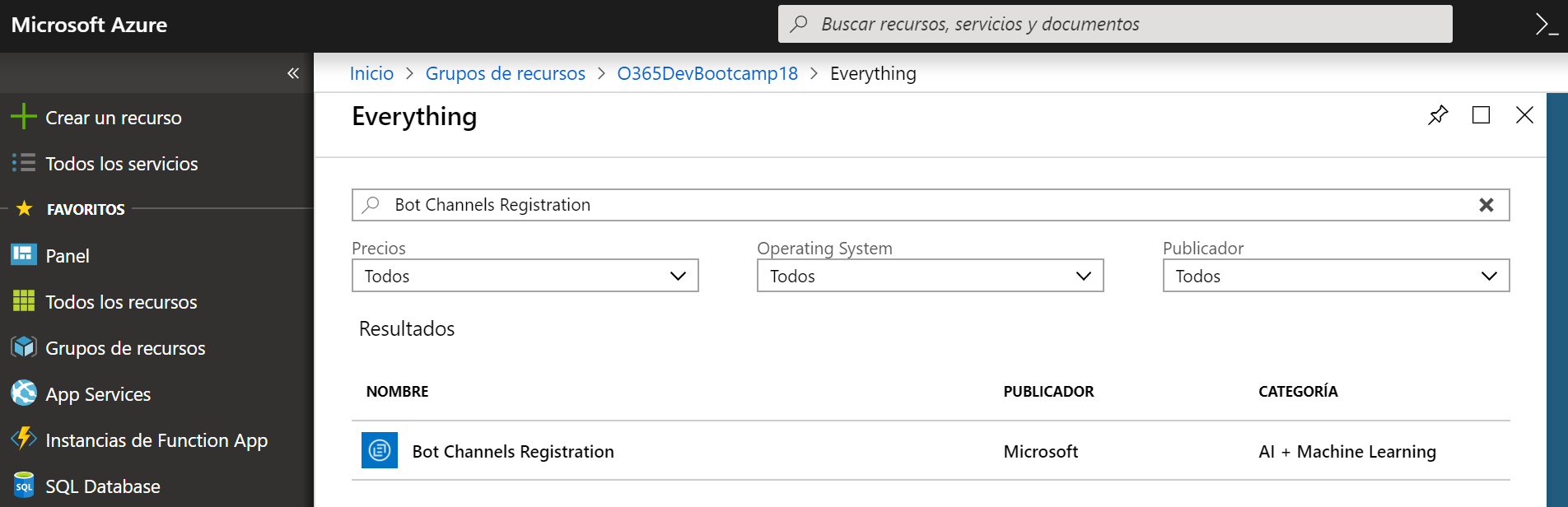


# Publicación del BOT desde Visual Studio

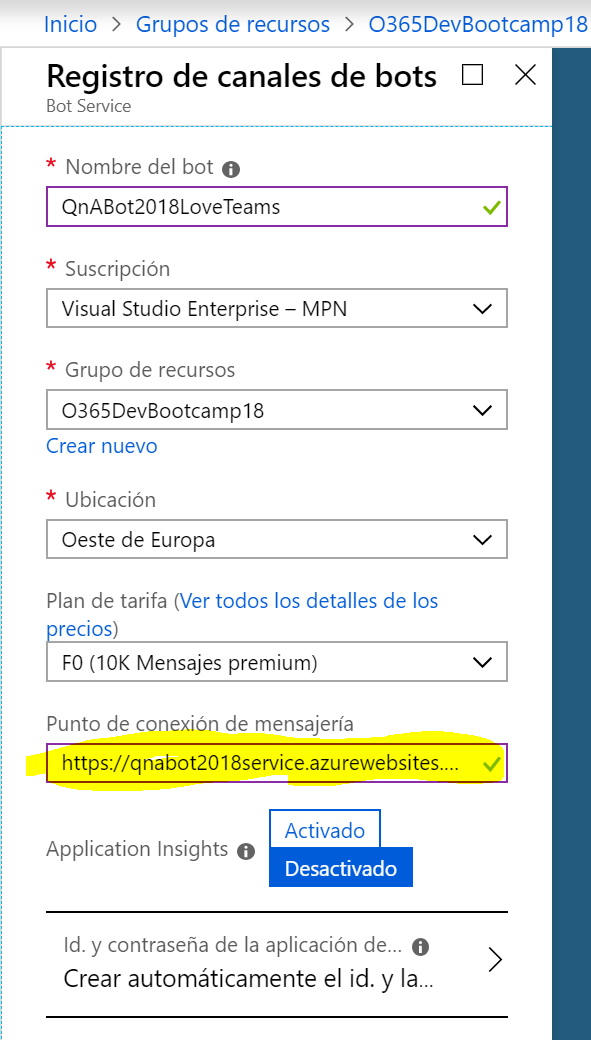
* Publicaremos ahora el servicio desde Visual studio:



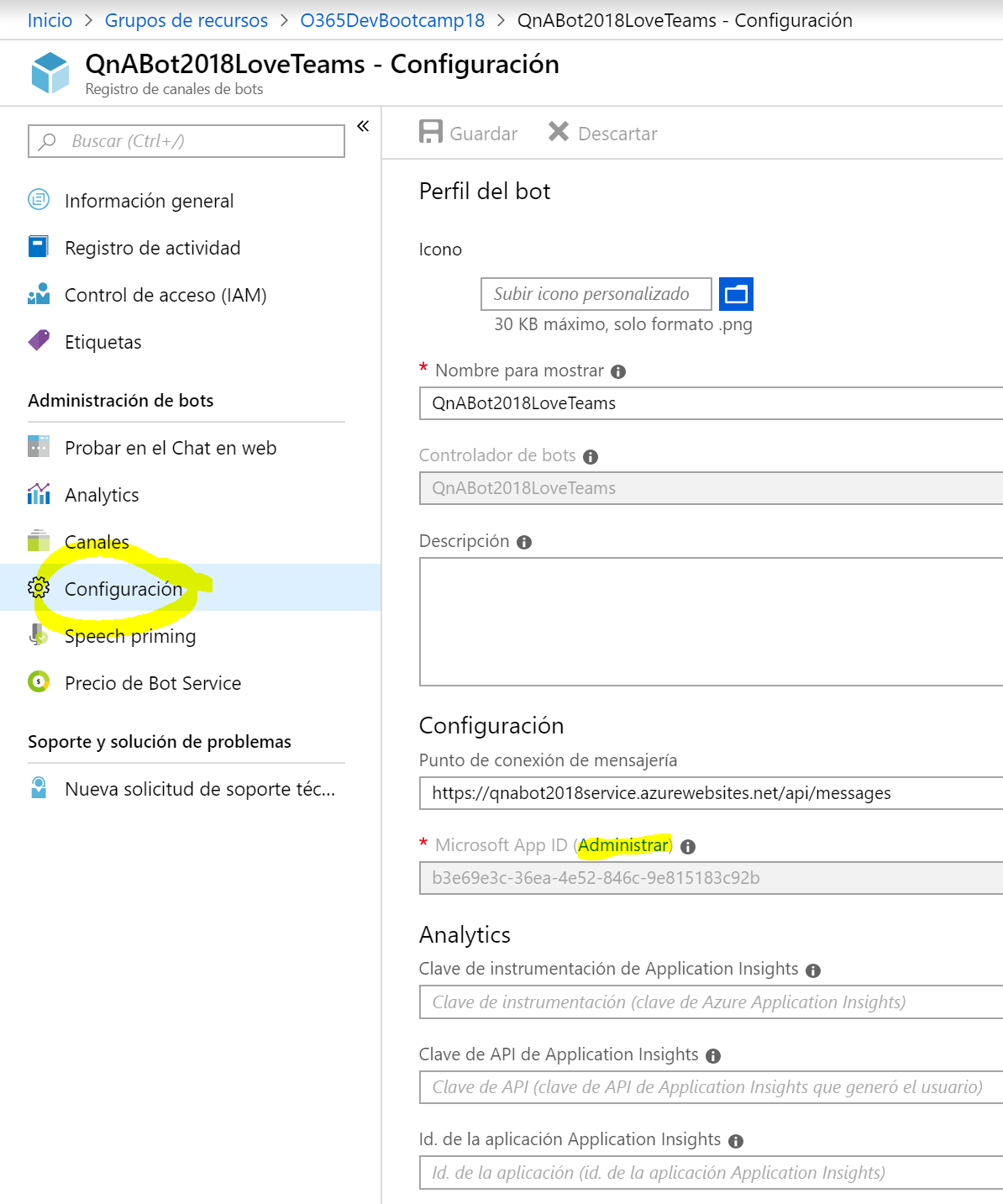
* Desde Microsoft Azure, accederemos al grupo de recursos donde alojamos el QaN y crearemos un “Bot channel registration”.



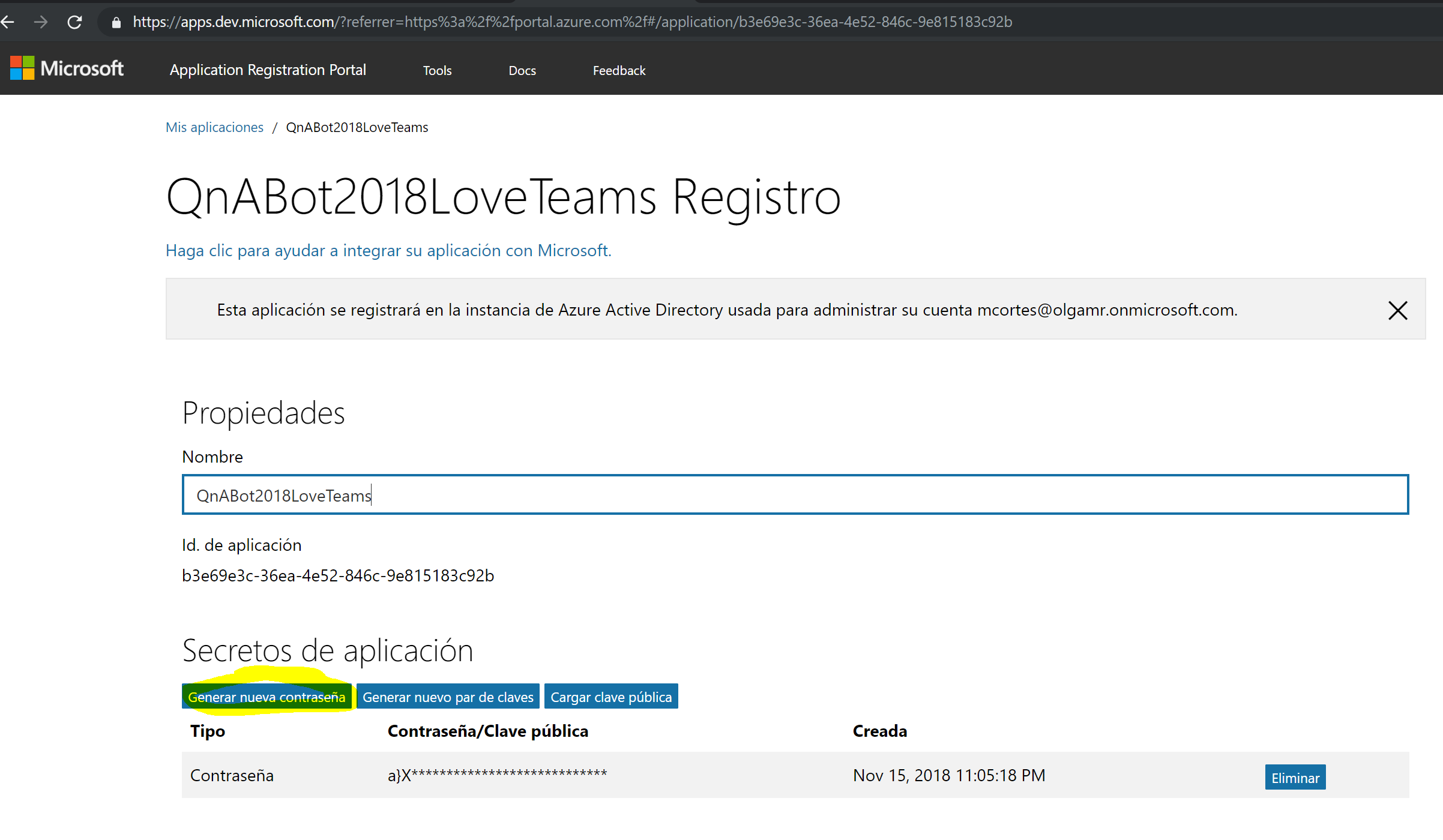
* En la pantalla de creación, indicaremos principalmente el “Punto de conexión de mensajería” donde indicaremos la url de entrada de mensajes del BOT publicado en Azure en el paso anterior. Para ello, copiaremos la url y añadiremos “api/messages”, quedando algo del estilo <https://qnabot2018service.azurewebsites.net/api/messages>

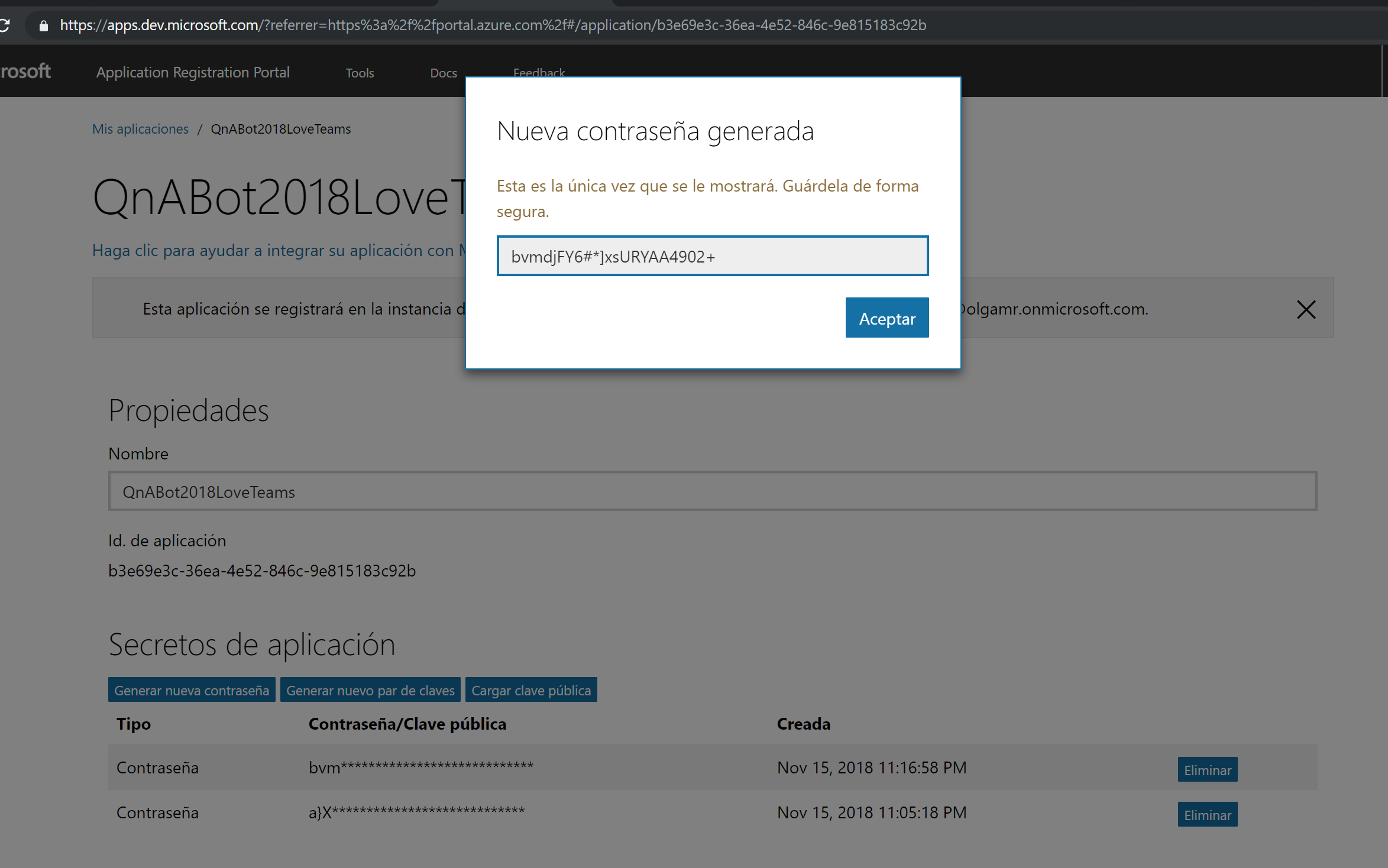


* Cuando haya terminado de crearse el canal, acceda al servicio y navegue hasta la opción “Configuración” donde habrá que copiar el nombre y el App Id para el registro en Teams.

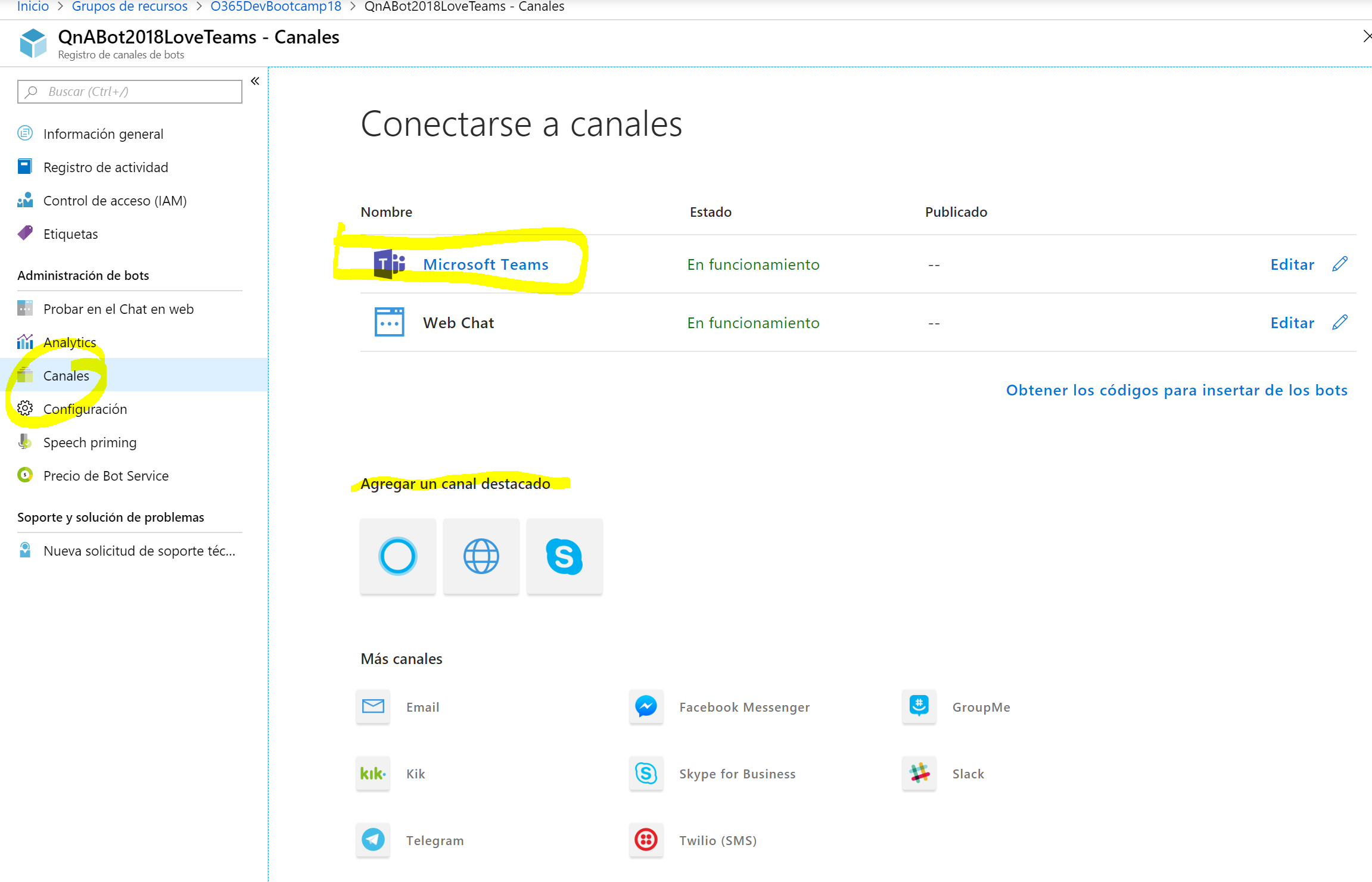


* Pulsaremos en “Administrar” y se abrirá una nueva página de registro de la aplicación de Azure, donde generaremos una nueva clave pulsando en “Generar nueva contraseña”. Copiaremos el valor para utilizarlo en el siguiente paso y guardaremos los cambios de la aplicación.





* Habilitaremos el canal para Teams desde la sección de Canales:



* Volveremos al proyecto de Visual Studio del bot y actualizaremos el fichero qnamaker.bot con los valores de AppID y AppPassword copiados en el paso anterior.

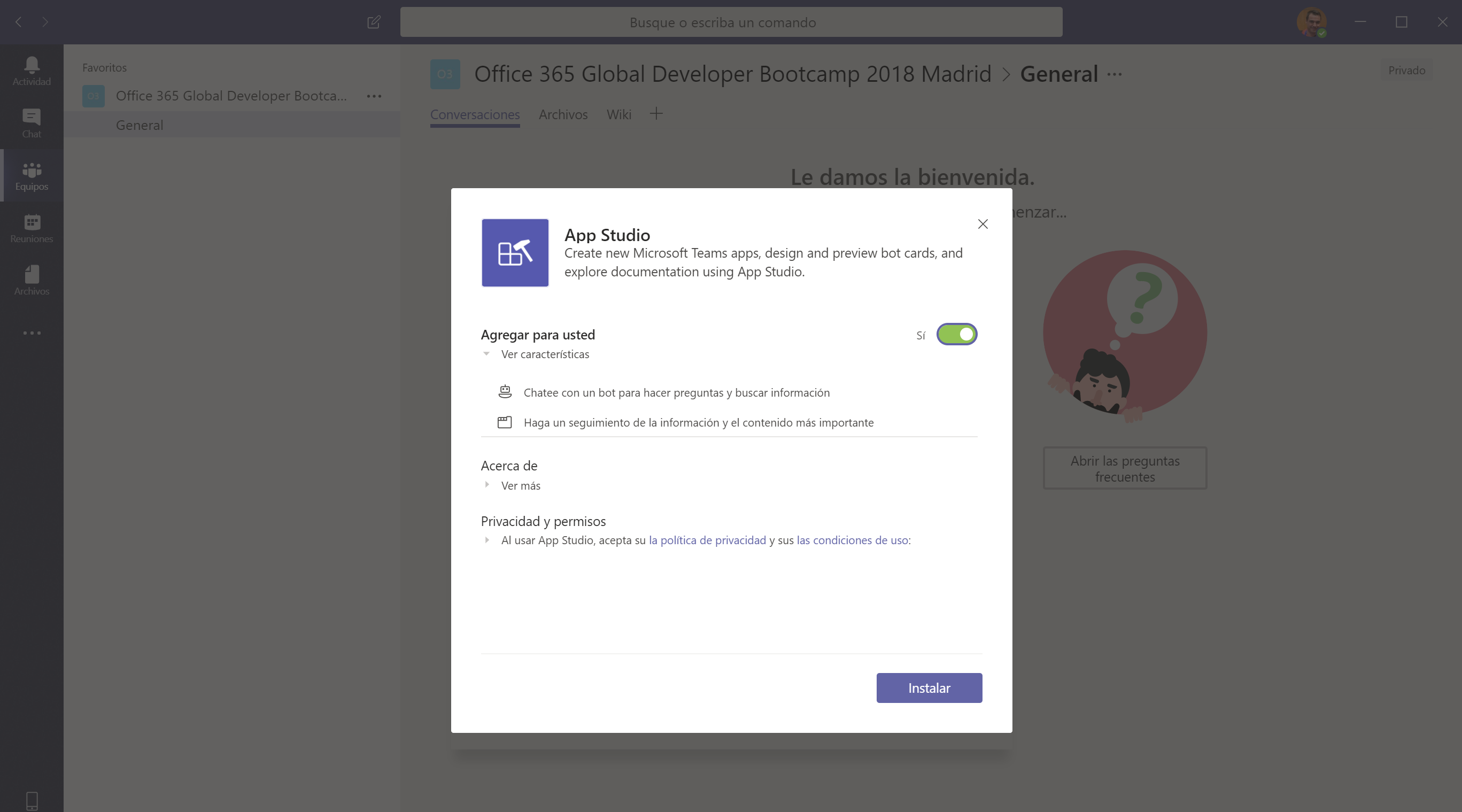


* Publicaremos de nuevo el proyecto con los nuevos valores.

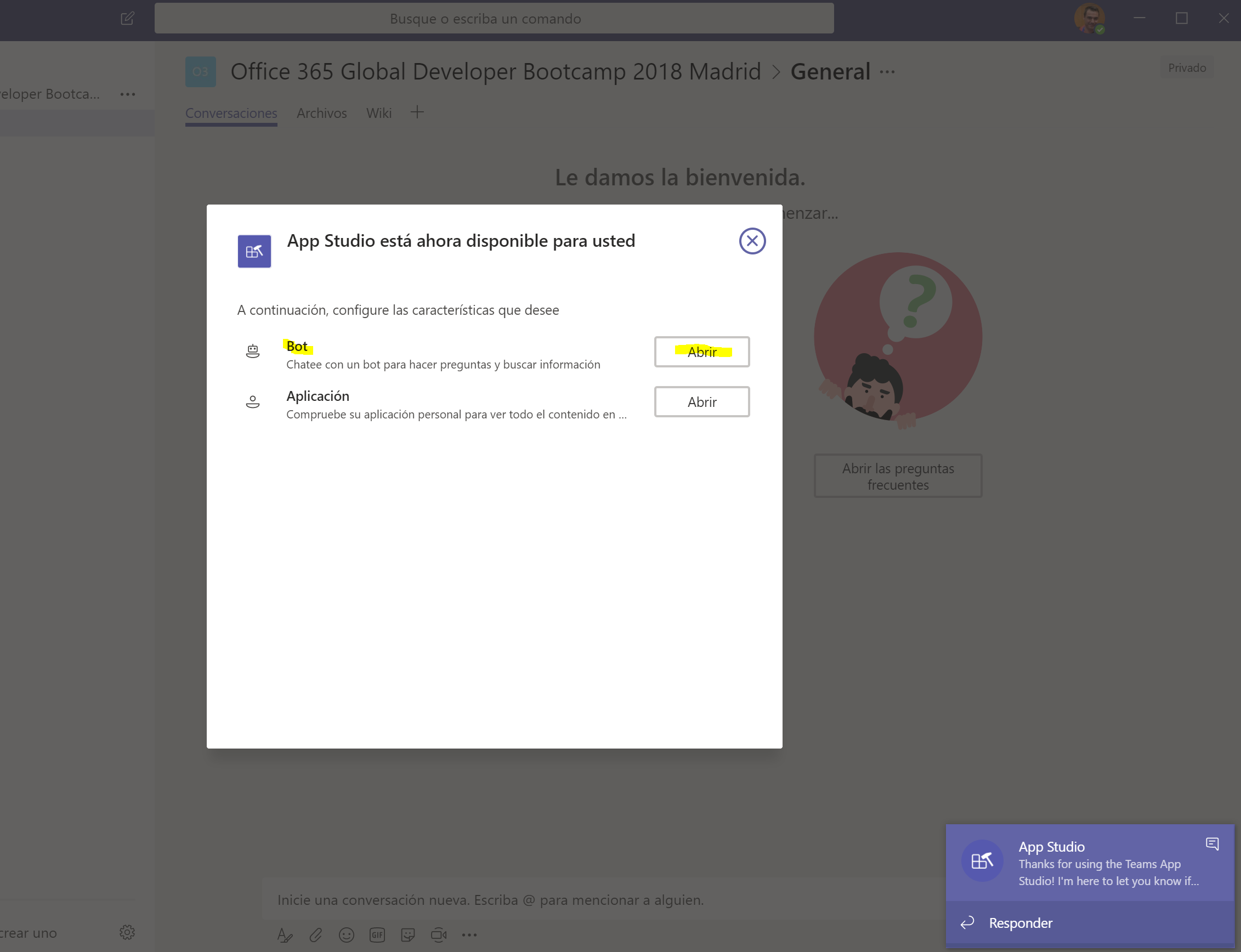
# Instalar App Studio

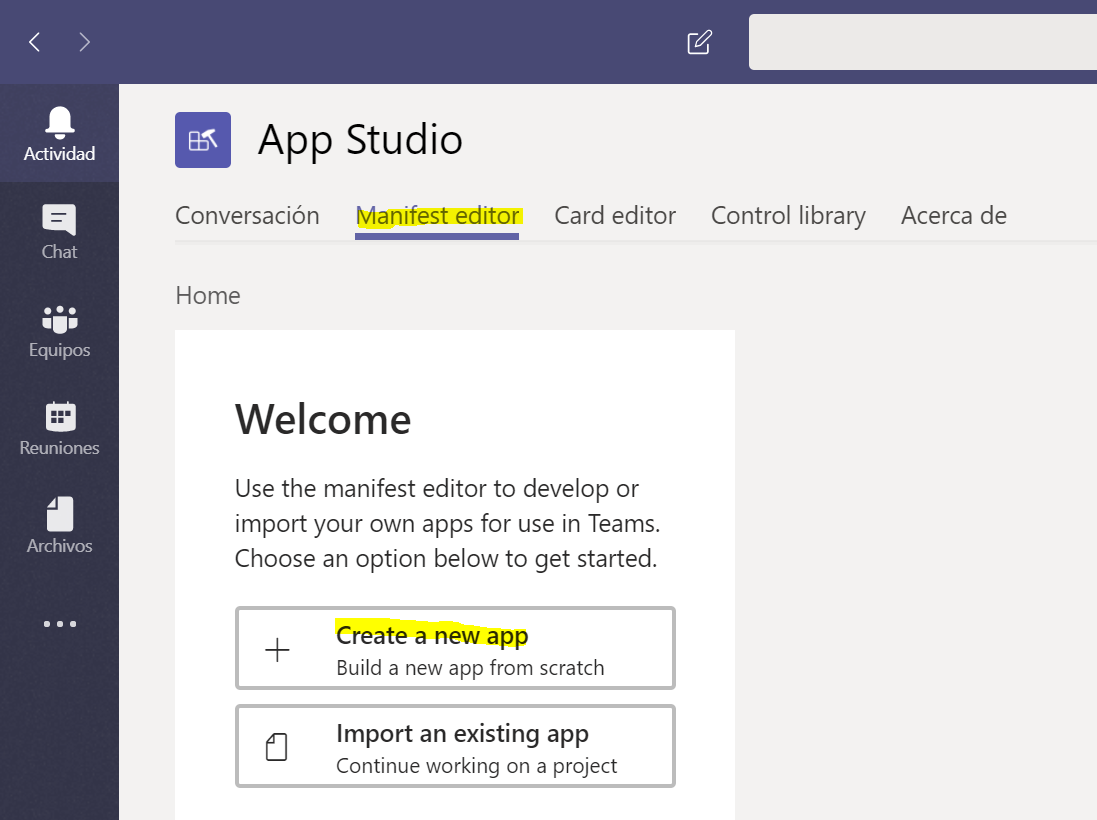
Utilzaremos [App Studio](https://docs.microsoft.com/en-us/microsoftteams/platform/get-started/get-started-app-studio) para registrar el bot en Teams.

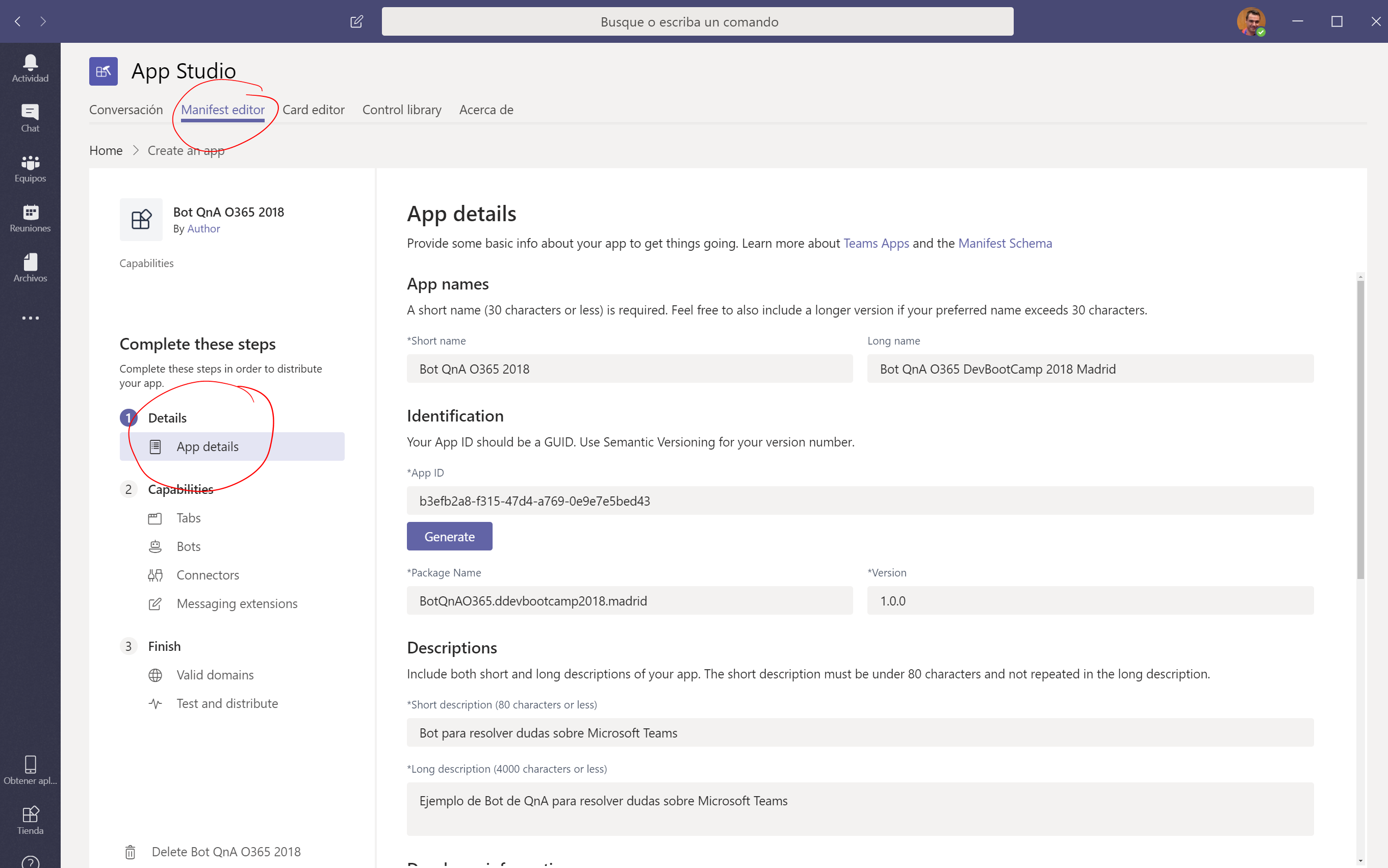
* Primero descargaremos e instalaremos App Studio desde el enlace <https://aka.ms/InstallTeamsAppStudio>.
* Al descargarse la aplicación, se abrirá el cliente Microsoft Teams con una ventana de diálogo para confirmar su registro. NOTA: Debe tener instalado previamente el cliente de Microsoft Teams en su máquina Windows.



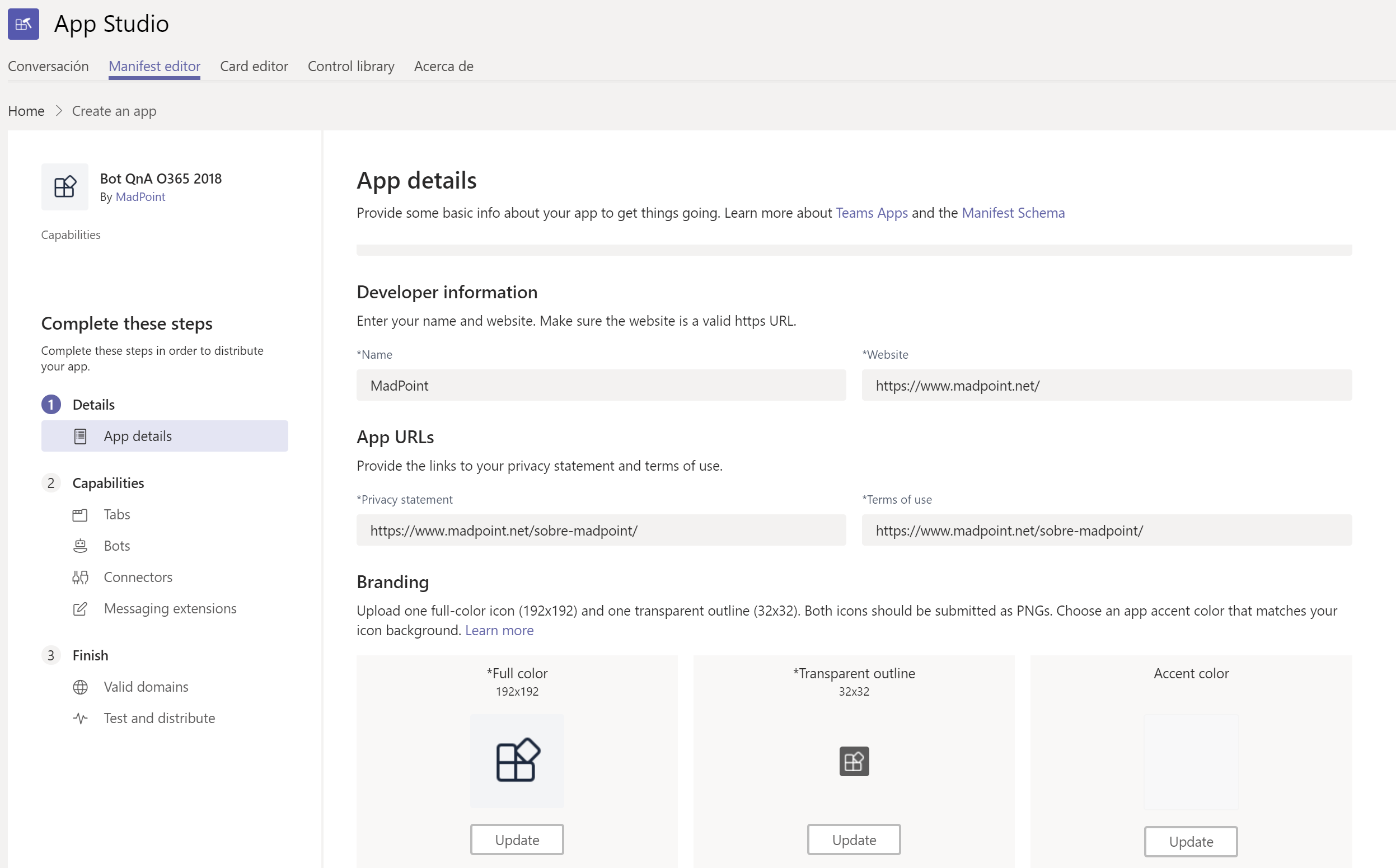
* Pulse en “Instalar” para finalizar la instalación de la aplicación App Studio en Microsoft Teams:



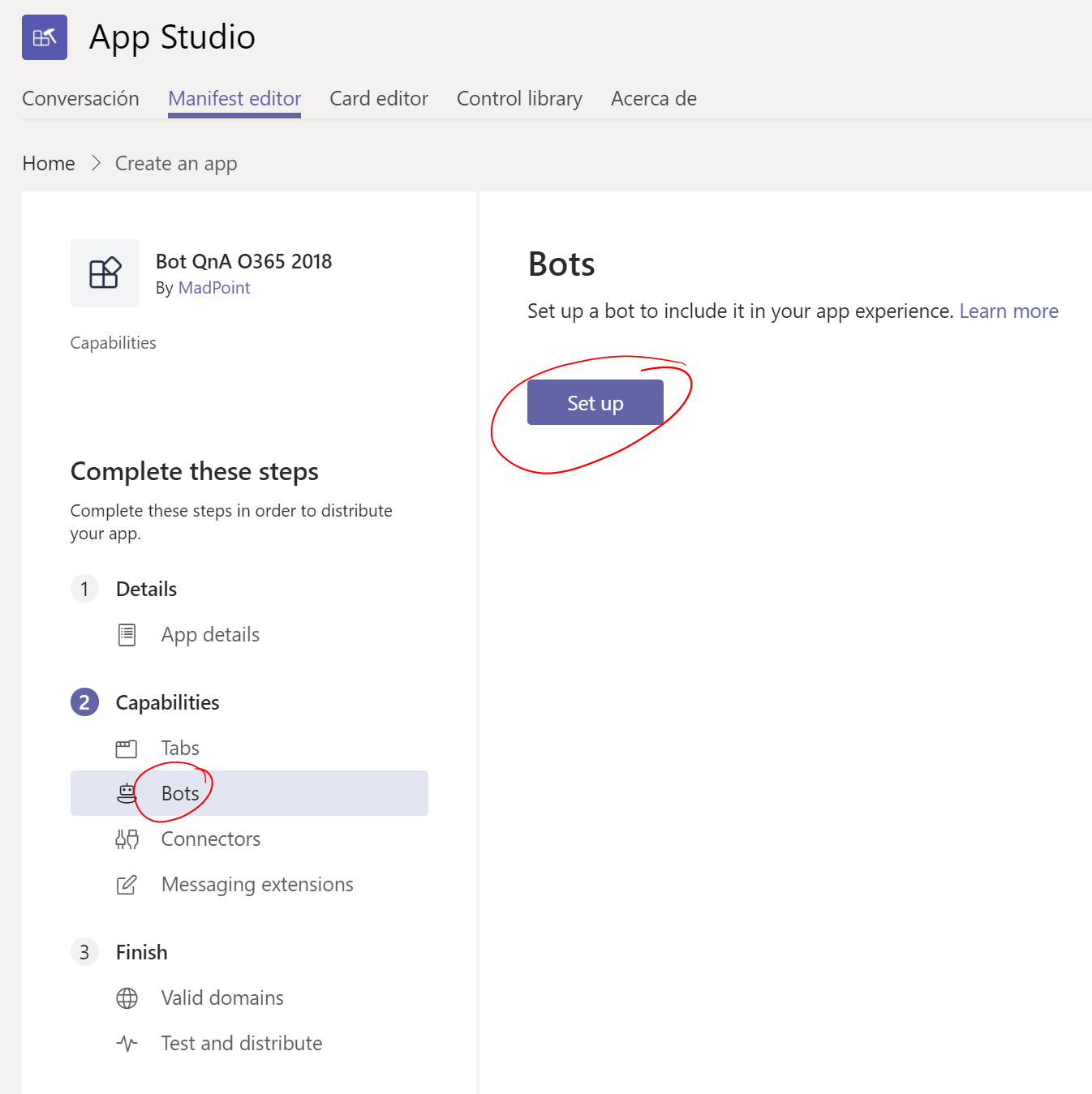
* Pulse en la pestaña de “Manifest Editor” y a continuación en “Create new app”:
* 
* Se abrirá entonces una pantalla desde la que tendremos que indicar los datos de nuestra aplicación y el tipo de aplicación. La pantalla estará dividida en diferentes secciones:
  + **Details**, desde la que indicaremos la información general de la aplicación como el título, datos de soporte, versión, identificador,…
  + **Capabilities**, donde seleccionaremos los elementos que dispondrá la aplicación, como una tab o un bot.
  + **Finish,** con un resumen de los datos introducidos.



* En App Details, introduciremos:
  + El título corto y un nombre largo.
  + El App ID, introduciendo a mano un GUID que identificará el App o bien pulsando en “Generate” para generar uno automáticamente.
  + Package name, con el nombre del fichero del paquete que se generará.
  + Versión, con el número de versión del app.
  + Una descripción corta y otra larga.
  + Informatción del desarrollador, con datos de la empresa y sitio público del fabricante.
  + App URLS, con las urls de las páginas de términos de uso y privacidad.

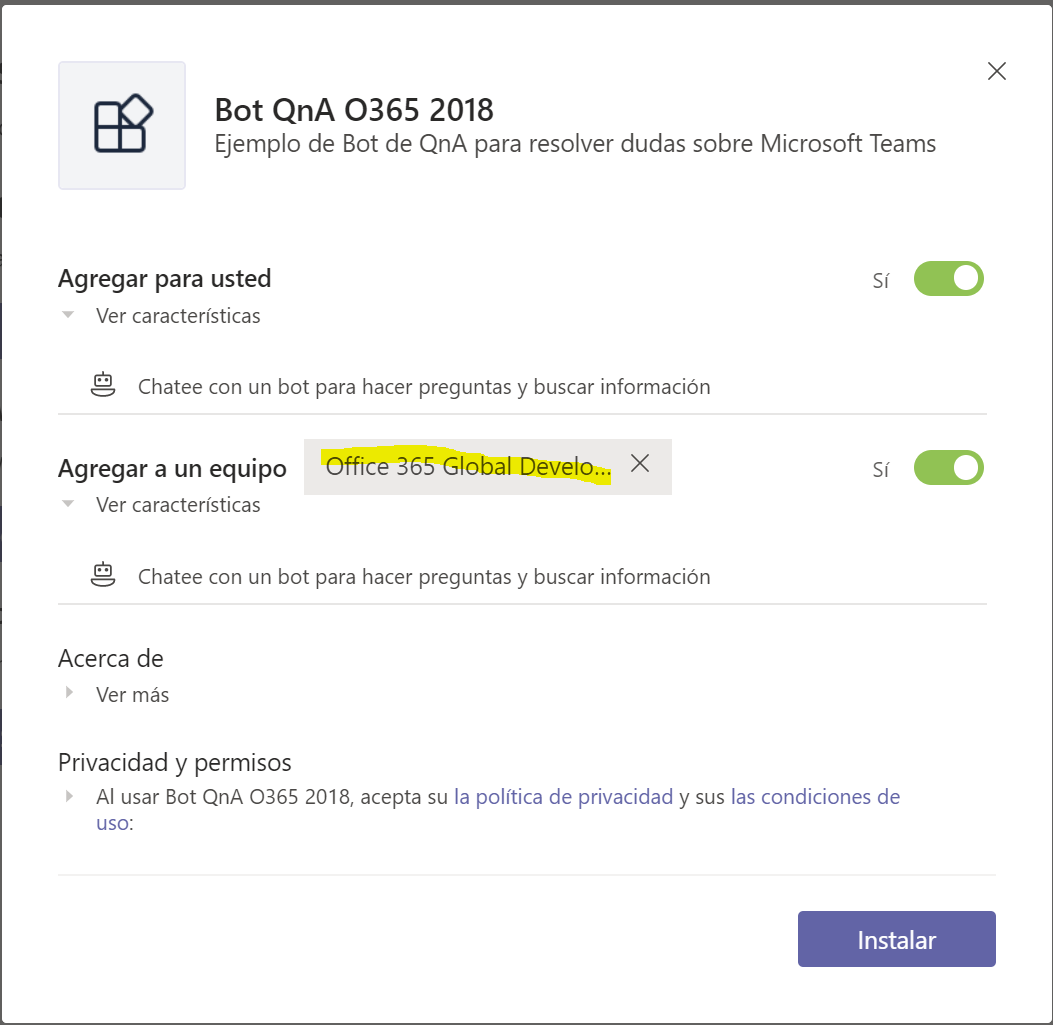


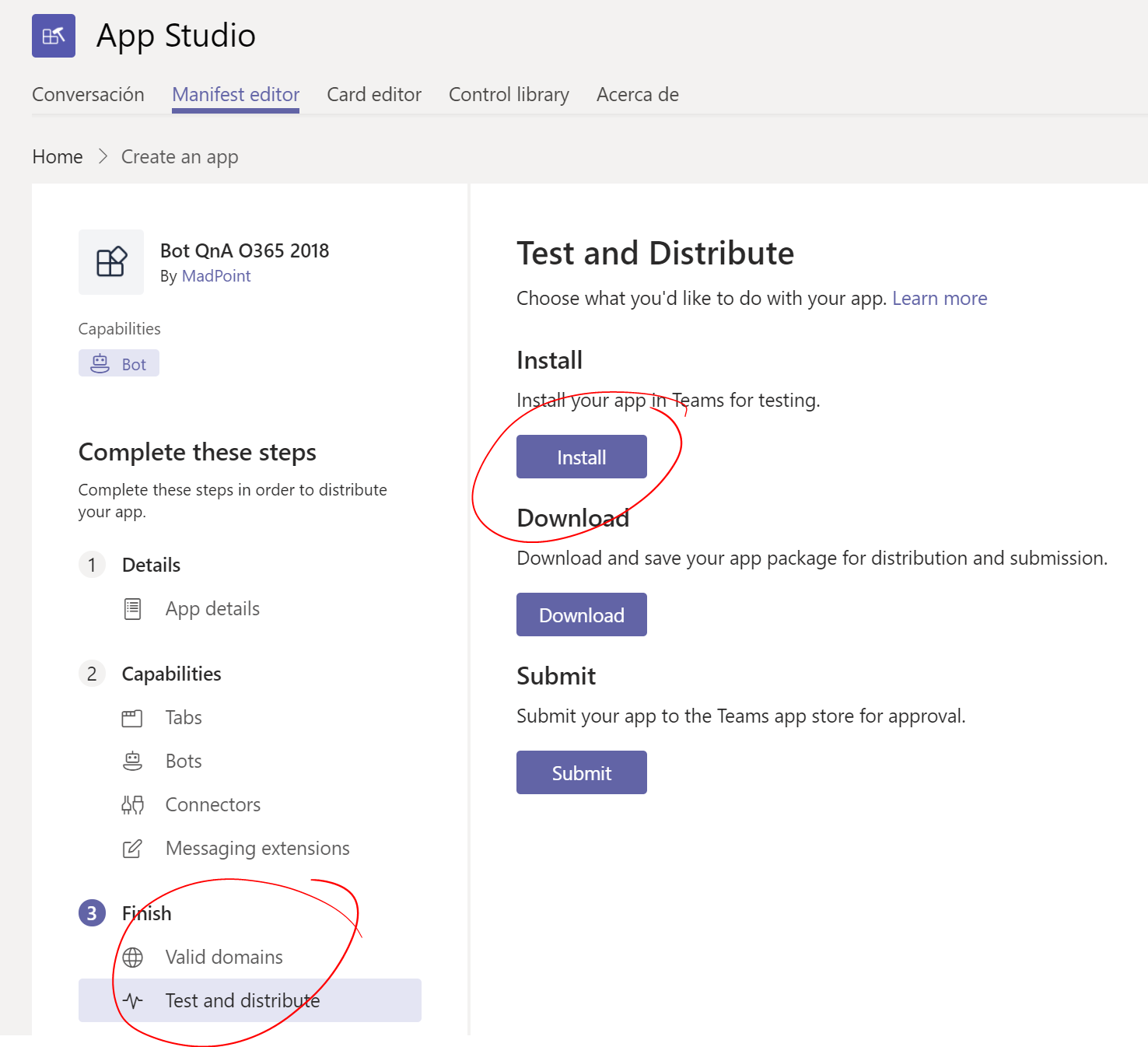
* En la sección de “Capabilities” seleccionaremos “Bot” y “Setup”, desde donde indicaremos los datos de nuestro bot previamente creado.



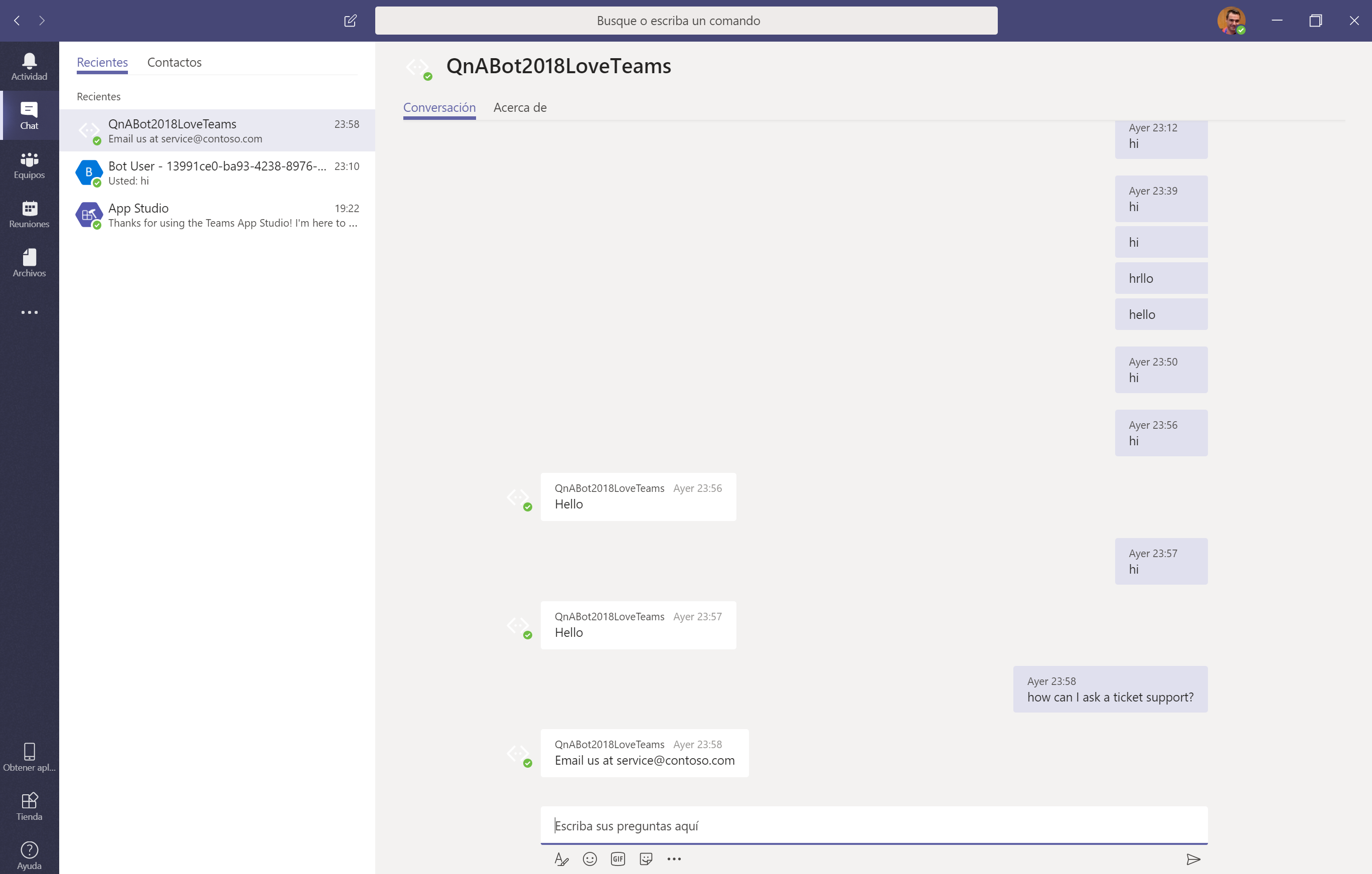
* En la ventana emergente de la pantalla de Setup, seleccionaremos la opción “Existin bot”:
  + El nombre del bot
  + El App Id
  + Las capacidades de envío de ficheros y mensajes.
  + El ámbito del bot, si será solo personal para conversaciones por chat o para Teams en un canal.



* Desde la sección 3, pulsaremos en “Test and distribute” y pulsaremos en “Install” apra instalar localmente la aplicación de Teams antes de distribuirla.
  + Selecciones el Team donde desea instalar la aplicación de prueba y pulse en “Instalar”.
* 



* Por último, probaremos el chat desde la sección de chats



# Referencias

Azure Bot Framework

<https://dev.botframework.com/>

What is QnA Maker?

<https://docs.microsoft.com/en-us/azure/cognitive-services/qnamaker/overview/overview>

Create a knowledge base

<https://docs.microsoft.com/en-us/azure/cognitive-services/QnAMaker/how-to/create-knowledge-base>

Add bots to Microsoft Teams apps

<https://docs.microsoft.com/en-us/microsoftteams/platform/concepts/bots/bots-overview>

BotBuilder Samples

<https://github.com/microsoft/botbuilder-samples>

Use QnA Maker to answer questions

<https://docs.microsoft.com/en-us/azure/bot-service/bot-builder-howto-qna?view=azure-bot-service-4.0&tabs=cs>

Quickly develop apps with App Studio for Microsoft Team

<https://docs.microsoft.com/en-us/microsoftteams/platform/get-started/get-started-app-studio>

<https://blogs.msdn.microsoft.com/geoffreyinnis/2017/10/19/bots-as-chat-agents-with-escalation-to-live-agents/>