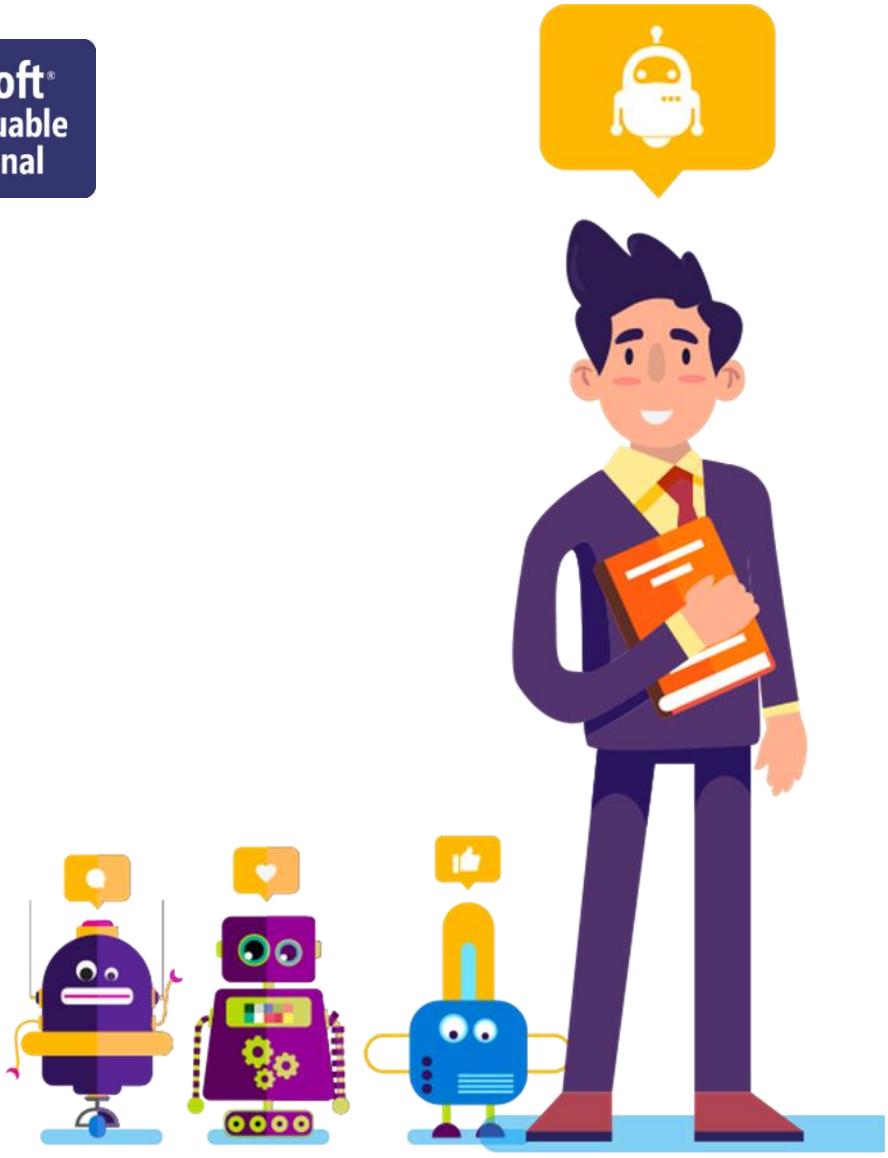
Diplomado Bots

Laboratorio

Versión: 1.0.0

Enero de 2018

Miguel Muñoz Serafín
@msmdotnet



Creando tu primer Bot con Bot Builder SDK for .NET

El ***Bot Builder SDK for .NET*** es un Framework fácil de utilizar que nos permite desarrollar bots utilizando Visual Studio y Windows. El SDK aprovecha C# para proporcionar una forma familiar para que los desarrolladores de .NET puedan crear bots poderosos.

En este ejercicio tendrás la oportunidad de crear un bot utilizando la plantilla ***Bot Application*** y el ***Bot Builder SDK for .NET***. Después de haber creado el bot, probarás su funcionamiento utilizando el ***Bot Framework Emulator***.

Ejercicio
Crear un bot con el Bot Builder SDK for .NET

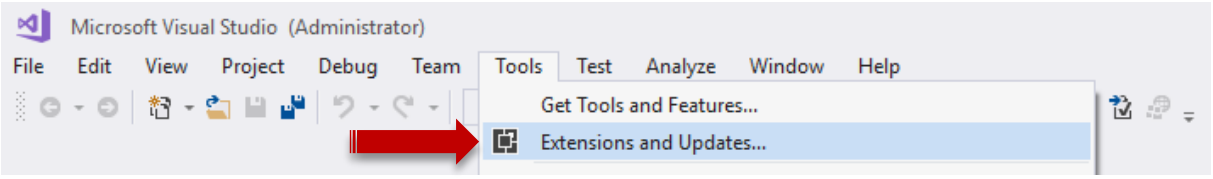
Realiza los siguientes pasos para configurar tu ambiente de desarrollo de bots con Visual Studio y crear tu primer Bot con C# y el ***Bot Builder SDK for .NET***.

Requisitos:

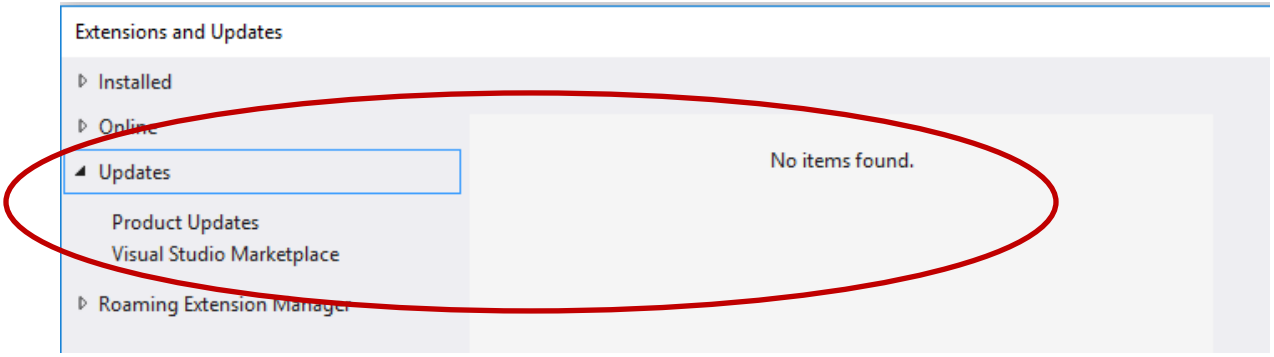
- **Visual Studio 2017**. Puedes utilizar la versión gratuita Visual Studio Community. El ***Bot Builder SDK for .NET*** actualmente soporta C#. ***Visual Studio for Mac*** no es soportado.

Preparar el entorno de desarrollo

1. Abre Visual Studio bajo el contexto de administrador.
2. Selecciona la opción ***Tools*** > ***Extensions and Updates…***



3. En la ventana ***Extensions and Updates*** verifica que no tengas actualizaciones pendientes. En caso de que haya actualizaciones disponibles, selecciona cada uno de los elementos y realiza su actualización.



Después de haber realizado todas las actualizaciones podrás continuar con los siguientes pasos para descargar las plantillas de proyectos de Visual Studio que te facilitarán el desarrollo de Bots.

4. Descarga el archivo "***Bot Application.zip"*** desde el siguiente enlace: http://aka.ms/bf-bc-vstemplate.

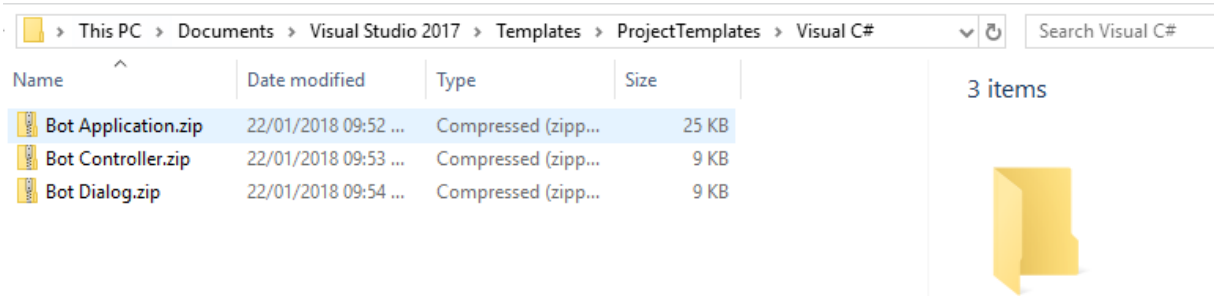
5. Descarga el archivo "***Bot Controller.zip***" desde el siguiente enlace: http://aka.ms/bf-bc-vscontrollertemplate.

6. Descarga el archivo "***Bot Dialog.zip***" desde el siguiente enlace http://aka.ms/bf-bc-vsdialogtemplate.

7. Copia los 3 archivos descargados hacia el directorio de plantillas de proyectos de Visual Studio para que estas plantillas se encuentren disponibles al momento de crear un nuevo proyecto con Visual Studio. Normalmente este directorio se encuentra en la siguiente ruta:

%USERPROFILE%\Documents\Visual Studio 2017\Templates\ProjectTemplates\Visual C#

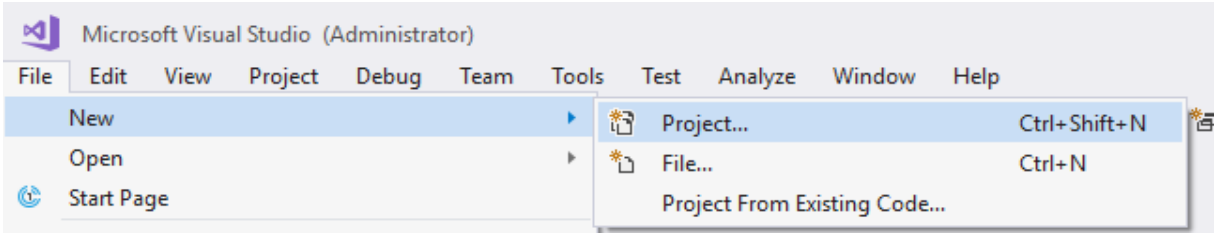
La siguiente imagen muestra los archivos de plantillas em el directorio correspondiente.



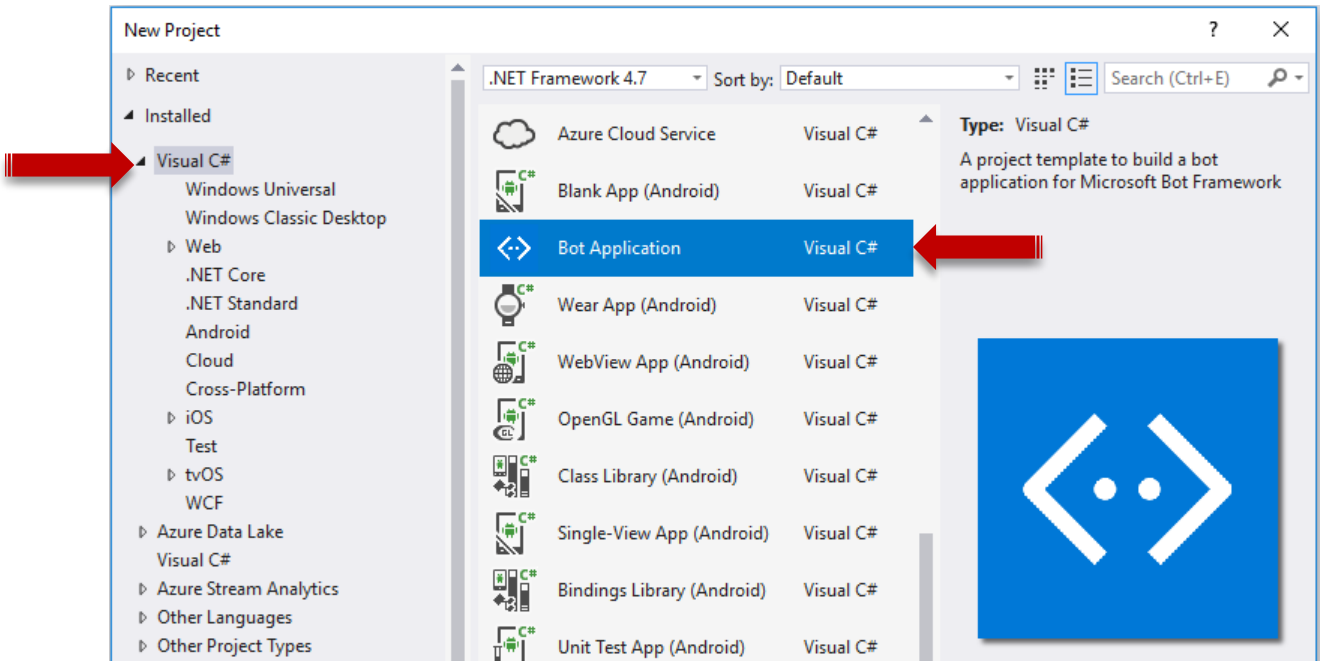
Crear el Bot

8. Abre Visual Studio bajo el contexto del administrador.

9. Selecciona la opción ***File*** > ***New*** > ***Project***.

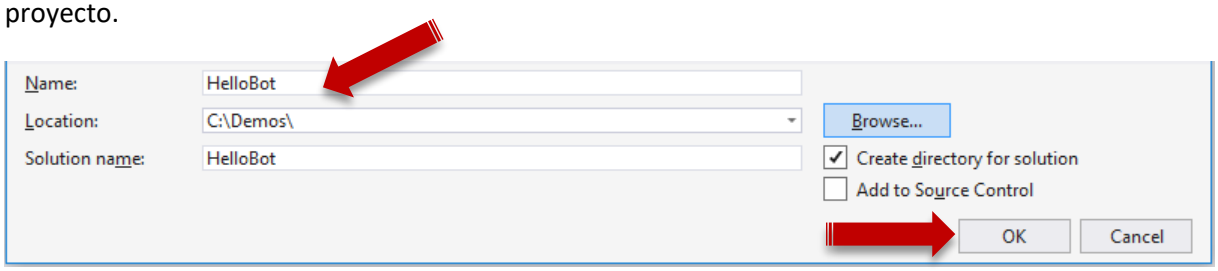


10. En la ventana ***New Project***, selecciona la plantilla ***Bot Application***.

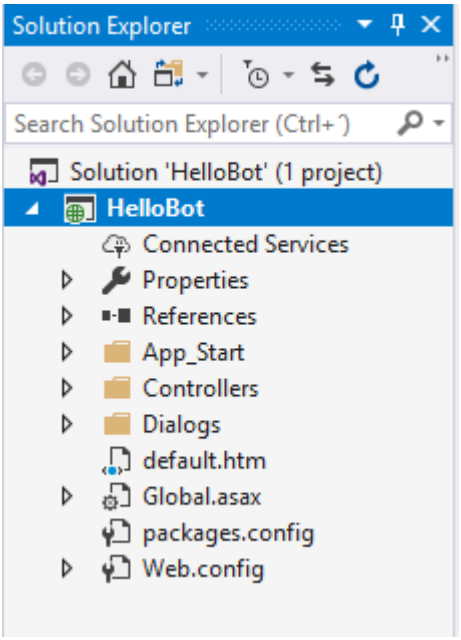


Al utilizar la plantilla ***Bot Application***, estarás creando un proyecto que contendrá todos los componentes requeridos para construir un bot sencillo, incluyendo una referencia al ***Bot Builder SDK for .NET***.

11. Asigna un nombre al proyecto, selecciona el directorio destino y haz clic en ***OK*** para crear el proyecto.

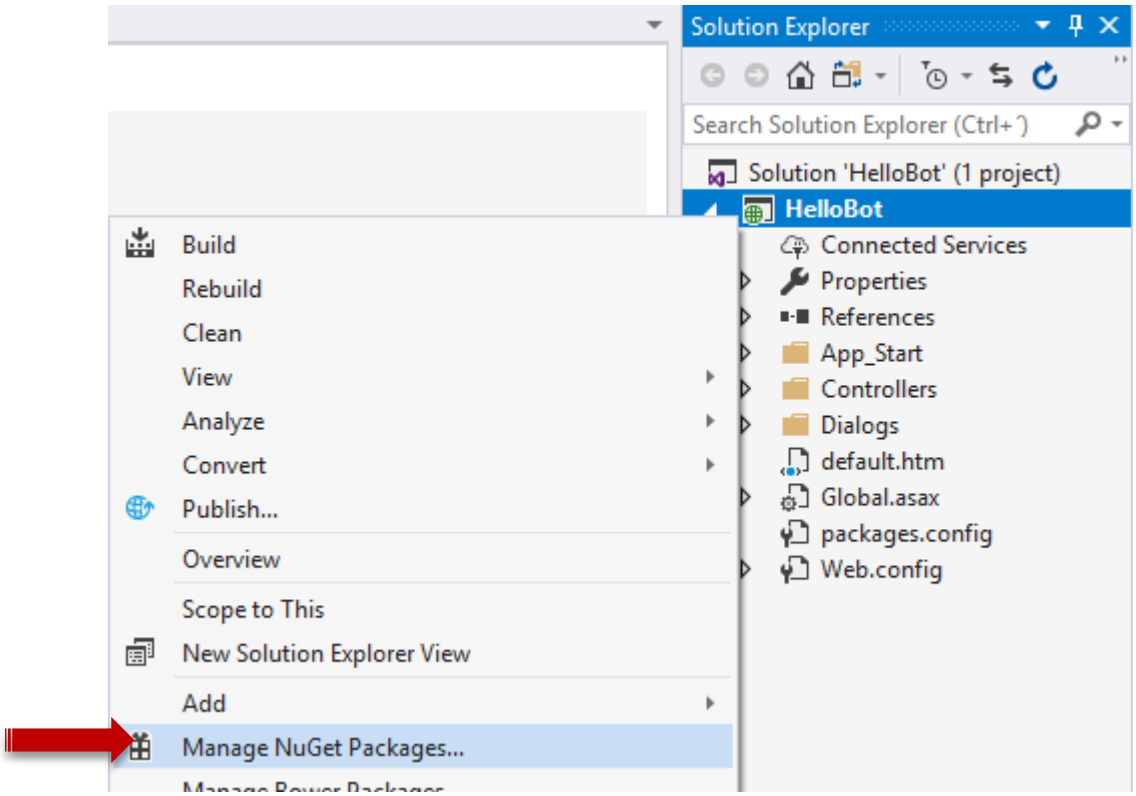


Al finalizar la creación del proyecto podrás ver la estructura de archivos similar a la siguiente.

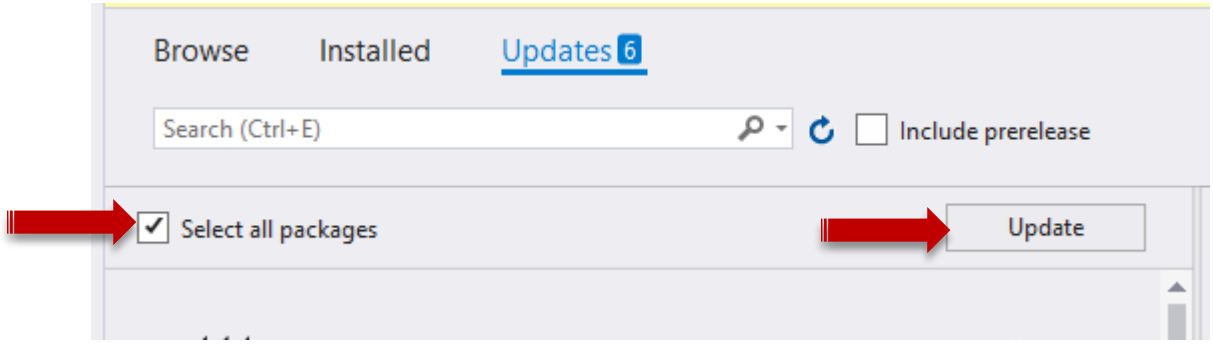


El siguiente paso será verificar que el proyecto haga referencia a la última versión del SDK.

12. Selecciona la opción ***Manage Nuget Packages…*** del menú contextual del proyecto.



13. Selecciona la casilla ***Select all packages*** y haz clic en ***Update*** para actualizar todos los paquetes.



Acepta la realización de cambios y acuerdos de licencia cuando te sea requerido.

Gracias a la plantilla ***Bot Application***, tu proyecto contiene ahora todo el código necesario para crear el bot. Para este ejercicio, no será necesario que agregues código adicional, sin embargo, antes de probar tu bot, echaremos una mirada a parte del código que la plantilla ***Bot Application*** proporcionó.

Explorar el código

14. Abre el archivo ***Controllers\MessagesController.cs***.

15. Examina el código del método ***Post***. El método ***Post*** recibe el mensaje del usuario e invoca a ***RootDialog***.

```
public async Task<HttpResponseMessage> Post([FromBody]Activity activity)
{
    if (activity.Type == ActivityTypes.Message)
    {
        await Conversation.SendAsync(activity, () => new Dialogs.RootDialog());
    }
    else
    {
        HandleSystemMessage(activity);
    }
    var response = Request.CreateResponse(HttpStatusCode.OK);
    return response;
}
```

El proceso de ***RootDialog*** procesa el mensaje y genera la respuesta. El método ***MessageReceivedAsync*** dentro del archivo ***Dialogs\RootDialog.cs*** envía una respuesta al usuario indicando el número de caracteres que tiene el texto que el usuario envió.

```
private async Task MessageReceivedAsync(IDialogContext context,
                                        IAwaitable<object> result)
{
    var activity = await result as Activity;
```

```
            // calculate something for us to return
            int length = (activity.Text ?? string.Empty).Length;

            // return our reply to the user
            await context.PostAsync(
                        $"You sent {activity.Text} which was {length} characters");

            context.Wait(MessageReceivedAsync);
        }
```

Probar el Bot

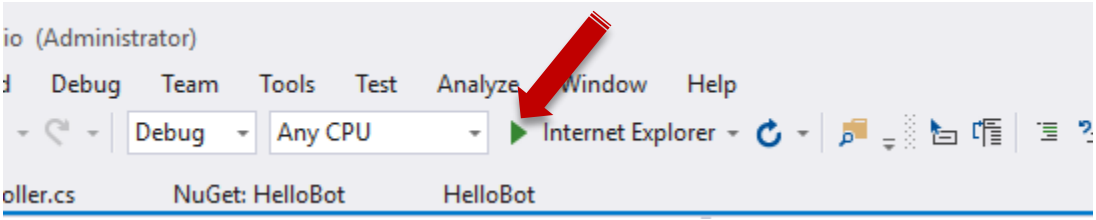
El siguiente paso será probar el funcionamiento del bot utilizando el ***Bot Framework Emulator***. El emulador es una aplicación de escritorio que nos permite probar y depurar nuestro bot ejecutándose en la computadora local (*localhost*) o de manera remota a través de un túnel.

16. Descarga el archivo ***botframework-emulator-Setup-3.5.35.exe*** desde el enlace https://emulator.botframework.com/.

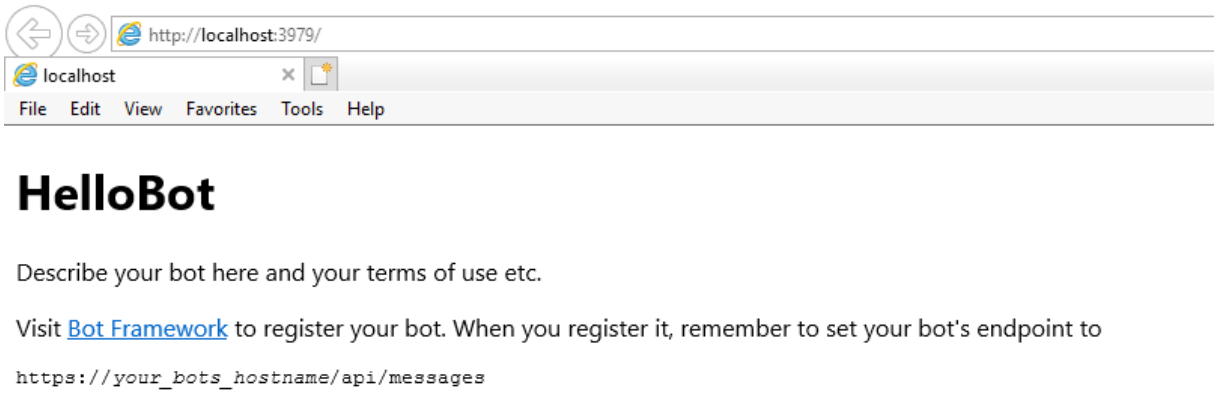
17. Ejecuta el archivo ***botframework-emulator-Setup-3.5.35.exe*** para instalar el emulador

Iniciar el bot

18. Ejecuta tu bot en Visual Studio utilizando un navegador web como anfitrión de la aplicación. La imagen siguiente muestra que el bot se lanzará en Internet Explorer cuando el botón de ejecución sea seleccionado.



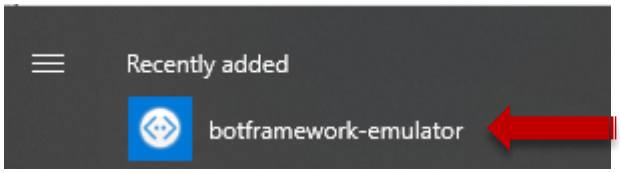
Al hacer clic en el botón de ejecución, Visual Studio compilará la aplicación, la desplegará hacia *localhost* y lanzará el explorador web para mostrar la página ***default.htm*** de la aplicación.



Iniciar el emulador y conectar tu bot

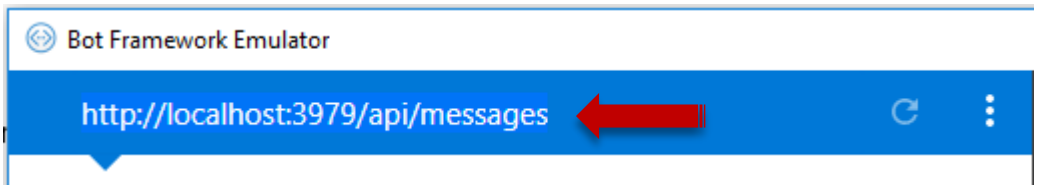
En este punto, tu bot se está ejecutando localmente. El siguiente paso será iniciar el emulador y conectarte a tu bot desde el emulador.

19. Abre el emulador.

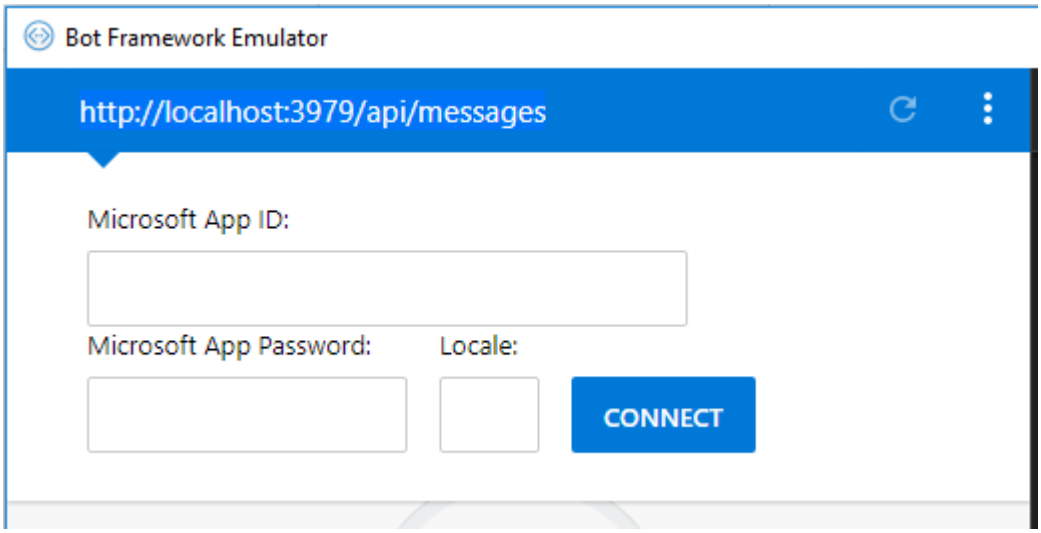


20. En la ventana ***Bot Framework Emulator***, escribe la siguiente dirección remplazando el puerto correcto donde se está ejecutando tu bot y que es mostrado por el navegador web.

http://localhost:puerto/api/messages



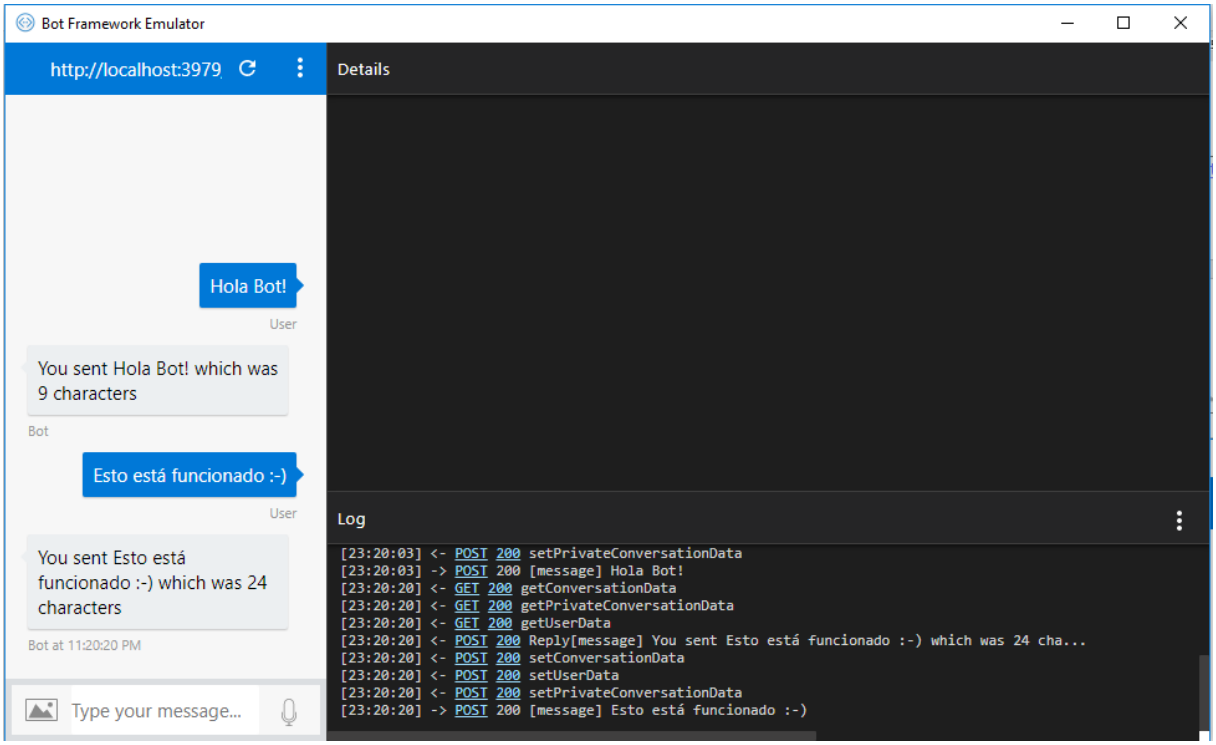
21. Haz clic en ***CONNECT***. No es necesario especificar **Microsoft App ID** ni **Microsoft App Password**. Esa información será necesaria solo cuando registres tu bot.



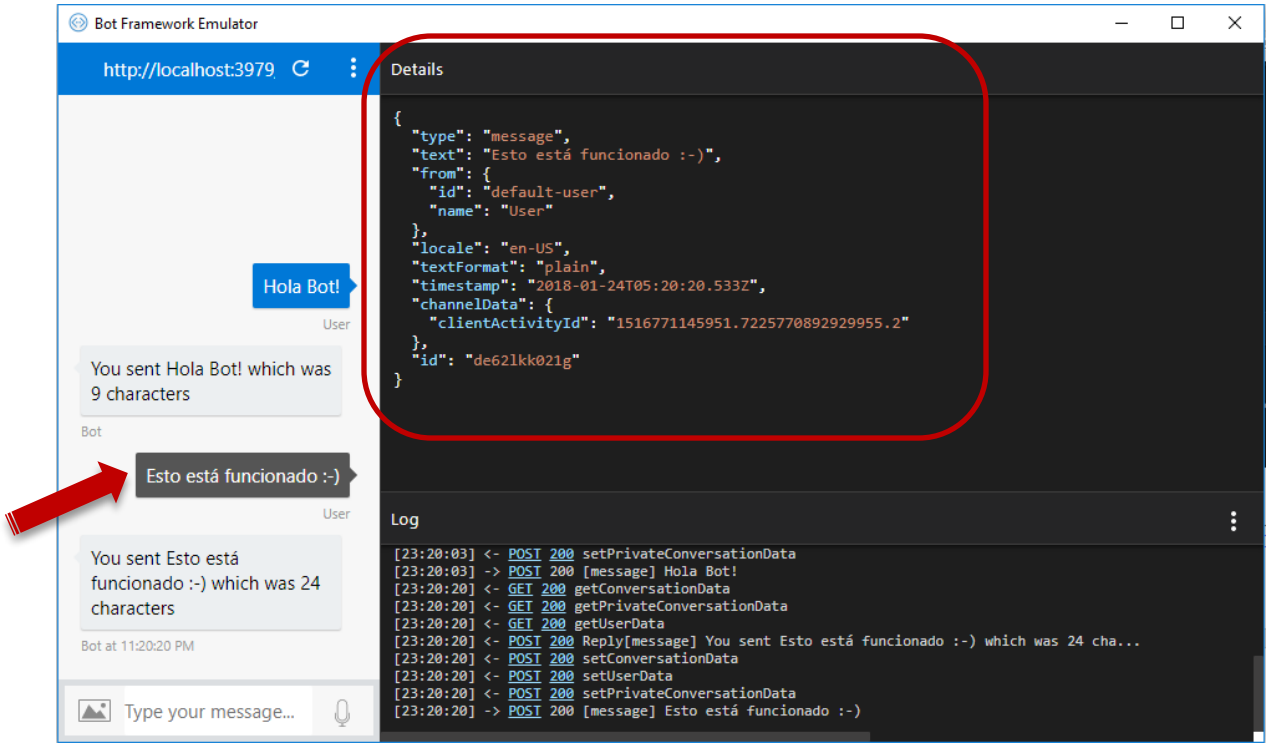
Probar el bot

Ahora que tu bot se está ejecutando localmente y está conectado al emulador, es momento de probar tu bot escribiendo algunos mensajes en el emulador.

22. Escribe algunos mensajes en el emulador. Podrás notar que el bot responde a cada uno de tus mensajes.



23. Haz clic en cualquier “burbuja” de dialogo. Podrás notar que el detalle del mensaje aparece en la ventana de detalle en formato JSON.



¡Con esto llegamos al final del ejercicio!

En este ejercicio creaste un bot sencillo utilizando la plantilla ***Bot Application*** y el B***ot Builder SDK for .NET***. Verificaste también la funcionalidad del bot utilizando el ***Bot Framework Emulator***.

¡Tu ambiente de desarrollo está listo para empezar a desarrollar bots!

Si lo deseas, puedes ver la versión en inglés de este laboratorio en el siguiente enlace:

Create a bot with the Bot Builder SDK for .NET
https://docs.microsoft.com/en-us/bot-framework/dotnet/bot-builder-dotnet-quickstart