



Traducción de voz en tiempo
real con Azure Cognitive
Services

BIENVENIDA

Obtenga información sobre cómo traducir y convertir voz en texto mediante la transcripción en tiempo real con Speech Translation API de Azure Cognitive Services.

En este módulo aprenderá a:

- Reconocer en qué consiste la traducción de voz
- Explorar Speech Translation API

2

Requisitos previos

Ninguno

Introducción **1 min**

Información general de conversión de voz en texto **3 min**

Ejercicio: Creación de un recurso de servicio de Voz **5 min**

Ejercicio: Visualización de las claves de suscripción y los puntos de conexión **5 min**

Ejercicio: Conversión de voz de un archivo de audio a texto **20 min**

Vista de idiomas de voz admitidos **4 min**

Ejercicio opcional: Escucha de datos de traducción entrantes **5 min**

Prueba de conocimientos **7 min**

Resumen **2 min**

INTRODUCCIÓN

El aspecto de conversión de voz en texto del servicio de Voz transcribe transmisiones de audio en texto. La aplicación puede mostrar este texto al usuario o actuar sobre él como entrada de comandos. Se puede usar este servicio con una biblioteca cliente del SDK (para las plataformas y los lenguajes admitidos), o bien con una API de transferencia de estado representacional (REST).

Con el servicio de Voz, se puede realizar lo siguiente:

- Ampliar el alcance de las aplicaciones en dispositivos móviles, de escritorio y en la web para proporcionar transcripción de voz a texto.
- Traducir con facilidad hacia y desde varios idiomas compatibles a través de la interfaz REST abierta del SDK de Voz. Esta API es un servicio automático de traducción de voz basado en la nube (también conocido como traducción automática).
- Realizar operaciones de texto a voz que puedan aceptar la entrada de texto y, después, generar una versión hablada de ese texto mediante voz sintetizada.

- Realización del reconocimiento de entidades mediante la integración con Language Understanding (LUIS)

Nota

Este módulo requiere una suscripción de Azure. Es posible que muchos de los servicios que se creen y usen sean gratuitos, pero necesitará una suscripción activa o una versión de prueba para completar los ejercicios. Si no tiene una suscripción a Azure, [cree una cuenta gratuita](#) antes de empezar.

Objetivos de aprendizaje

Objetivos de este módulo:

- Obtener información sobre algunas de las capacidades de conversión de voz en texto.
- Explorar el servicio de conversión de voz en texto para convertir un archivo de audio en una salida de texto.
- Explorar la capacidad de convertir la voz de un micrófono a una salida de texto.

INFORMACIÓN GENERAL DE CONVERSIÓN DE VOZ EN TEXTO

El **aspecto** de conversión de voz en texto de los servicios de Voz, en Azure Cognitive Services, proporciona transcripción en tiempo real de transmisiones de audio basada en inteligencia artificial y aprendizaje automático. Las Speech Services API permiten a los desarrolladores agregar transcripciones de voz de un extremo a otro en tiempo real a sus aplicaciones o servicios.

6

Los servicios de conversión de voz en texto se exponen a través de [API basadas en REST](#) independientes de la plataforma o del [SDK de Voz](#). Estas API permiten integrar estos servicios en cualquier solución que requiera traducción de voz de varios idiomas. La forma en que se tiene acceso a estos servicios variará en función de si se usa la API REST o el SDK de Voz. Los ejercicios de este módulo utilizarán el SDK de Voz.

Los servicios de Voz están diseñados para realizar conversión de voz en texto en tiempo real en escenarios como los siguientes:

- Traducción de presentaciones en directo
- Comunicaciones traducidas en persona o remotas
- Asistencia al cliente
- Inteligencia empresarial

- Subtitulado de elementos multimedia
- Interacciones con inteligencia artificial multilingüe

Nota

Es posible que conozca un servicio similar de Azure: Translator Speech API. Translator Speech es una API heredada que se va a reemplazar por la nueva Speech Translation API. Translator Speech API estará disponible hasta octubre de 2019. Obtenga más información sobre cómo [migrar de Translator Speech API a Speech Translation API](#) antes de que se deshabilite. Tenga en cuenta también que, en caso de que lo haya estado usando, a partir del 15 de octubre de 2019 Bing Speech ya no está disponible. Vea las guías de migración en docs.microsoft.com para mover sus soluciones a los servicios de Voz.

7

De forma predeterminada, el servicio de conversión de voz en texto usa el modelo de lenguaje universal. Este modelo se ha entrenado con datos propiedad de Microsoft y se implementa en la nube. Es óptimo para escenarios de conversación y dictado. Al usar conversión de voz en texto para el reconocimiento y la transcripción en un entorno único, se pueden crear y entrenar modelos acústicos, de idioma y de pronunciación personalizados. La personalización es útil para abordar el ruido ambiente o el vocabulario específico del sector.

Este módulo no tratará la personalización. Es posible que los módulos futuros traten este asunto. También puede visitar la página de [Voz personalizada](#) para obtener más información.

EJERCICIO: CREACIÓN DE UN RECURSO DE SERVICIO DE VOZ

Antes de empezar a realizar la traducción de voz a texto, debe crear un recurso de voz de Azure. Puede hacerlo mediante Azure Portal, la CLI de Azure o Cloud Shell. En este ejercicio se usará Azure Portal.

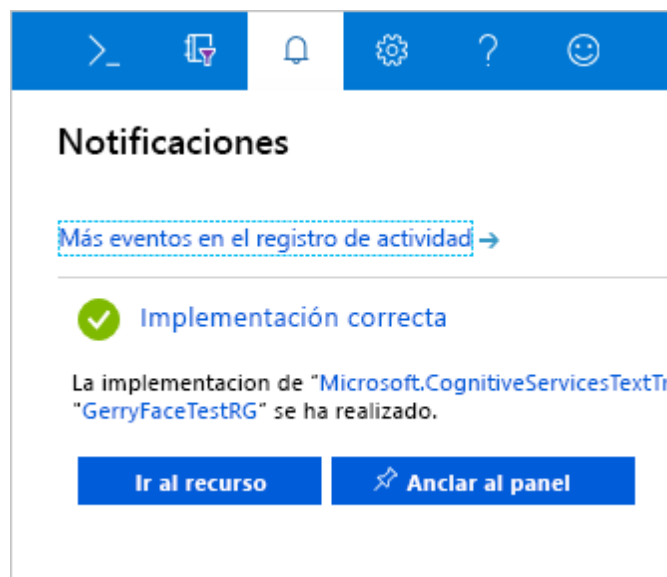
1. Inicie sesión en [Azure Portal](#).
2. Haga clic en **+ Crear un recurso**. En el cuadro de texto **Buscar en Marketplace**, escriba **voz** y presione Entrar.
3. En la lista **Resultados**, seleccione **Voz**. En el panel de **Voz**, seleccione **Crear**.
4. Escriba un nombre único para el recurso de servicio de Voz.
5. Seleccione una suscripción adecuada.
6. Elija una ubicación para hospedar el recurso. Esta suele ser la región donde se utilizará el recurso.
7. Como **Plan de tarifa**, seleccione un nivel de servicio. Los niveles de servicio pueden cambiar, pero actualmente puede seleccionar F0 o S0. Para las pruebas, hemos seleccionado F0.

8. Cree un grupo de recursos denominado **mslearn-speechapi** para almacenar los recursos. También puede elegir un grupo de recursos existente si lo desea.
9. Haga clic en **Crear** para crear una suscripción a Speech Translation API.

Tras una breve demora, la nueva suscripción de Speech Translation API estará disponible, y se generarán claves de API nuevas para su uso mediante programación.

Sugerencia

Si se pierde la notificación que indica que el recurso se ha publicado, haga clic en el icono de notificación de la barra superior del portal y, después, haga clic en **Ir al recurso**, como se muestra en la imagen siguiente:



Después de crear un recurso de servicio de Voz, ya se puede acceder al punto de conexión y las claves de suscripción de la API.

EJERCICIO: VISUALIZACIÓN DE LAS CLAVES DE SUSCRIPCIÓN Y LOS PUNTOS DE CONEXIÓN

Para acceder al servicio de Voz desde una aplicación, necesita obtener dos datos en Azure Portal:

- Una *clave de suscripción*, que se pasa con cada solicitud para autenticar la llamada.
- El *punto de conexión* que expone el servicio en la red.

Se necesitará la clave de suscripción al usar el SDK de Voz o las API REST, pero el punto de conexión solo será necesario para el acceso a la API REST. El uso del SDK de Voz en una aplicación utiliza la clave, pero también requiere una región. Verá el uso de la información de la clave y la región en el ejercicio más adelante en el módulo.

Visualización de las claves de suscripción y el punto de conexión

1. En el menú de la izquierda del portal, seleccione **Grupos de recursos** y después el grupo de recursos **mslearn-speechapi** que ha creado para Speech Translation API.

2. Seleccione el nombre de la suscripción de Speech Translation API (por ejemplo, **SpeechTranslation**).
3. Hay dos ubicaciones que mostrarán el punto de conexión y, por lo menos, una clave.
4. En el grupo **ADMINISTRACIÓN DE RECURSOS**, seleccione la opción **Inicio rápido** para mostrar valores de **Clave1** y **Punto de conexión**.
5. La segunda opción se usa para ver una segunda clave del servicio. En el grupo **ADMINISTRACIÓN DE RECURSOS**, seleccione **Claves y punto de conexión** para ver el nombre del servicio, el punto de conexión y dos claves de API.
6. Copie el valor de **CLAVE 1** o **CLAVE 2** en el Portapapeles para usarlo en una aplicación.

EJERCICIO: CONVERSIÓN DE VOZ DE UN ARCHIVO DE AUDIO A TEXTO

En este ejercicio, usará el servicio de Voz que se ha creado anteriormente. En este ejercicio no se requiere ninguna configuración en el equipo local para un entorno de desarrollo, ya que se utilizará Visual Studio Code en línea. En los pasos se detallará lo que se necesita para que el entorno en línea esté configurado correctamente y las tareas que serán necesarias para que cada ejercicio funcione correctamente. Todos los recursos se almacenan en un repositorio de GitHub al que se vinculará durante los pasos del ejercicio.

12

1. Para empezar, abra una pestaña o ventana del explorador y vaya a [Visual Studio Code en línea](#).

[!NOTE] En este momento, Safari no es un explorador compatible con la versión preliminar de VS Code en línea.

2. Necesitará una suscripción de Azure para crear un entorno para Visual Studio Code en línea.
3. Seleccione el botón **Primeros pasos**.
4. Inicie sesión con la cuenta de Microsoft que esté vinculada a su suscripción de Azure.

5. Si es la primera vez que usa Visual Studio Code en línea, se le pedirá que cree un plan nuevo, el cual se debe seleccionar en la lista desplegable de la parte superior de la página.
6. Seleccione **Crear entorno**.
7. Seleccione la Suscripción en el panel **Seleccionar facturación** y elija una ubicación para el servicio.
8. Seleccione **Crear**.
9. Una vez que se haya creado el plan, se podrá crear el primer entorno.
10. Si el panel **Crear entorno** no se abre automáticamente, seleccione **Crear entorno**.
11. Escriba el nombre del entorno que prefiera, por ejemplo, **SpeechToTextCS**.
12. Copie la siguiente dirección URL y péguela en el área de repositorio de Git en el panel **Crear entorno**, <https://github.com/MicrosoftLearning/SpeechToTextCS>.
13. Deje el resto de opciones con sus valores predeterminados y seleccione **Crear**.
14. El entorno tardará unos minutos en crearse, así que, mientras espera, vaya al servicio Voz en Azure Portal y copie una de sus claves y anote también la región en la que se ha creado el servicio.

15. Una vez que el entorno indica que está disponible, seleccione el título o los tres puntos verticales en la parte inferior derecha del cuadro de entorno y elija **Conectar**.
16. Usaremos C# y .Net Core para este ejercicio, así que, una vez que se abra la ventana de VS Code, seleccione el menú de hamburguesa, **Ver y Terminal** para iniciar el terminal.
17. Cuando empiece a escribir un comando, recibirá un mensaje que indica que el terminal es de solo lectura. Seleccione el botón para abrir un terminal nuevo.
18. Compruebe que **Bash** está seleccionado en la lista desplegable de la derecha Seleccionar shell predeterminado.
19. Escriba el comando `dotnet new console` y presione Entrar.
20. Se configura una nueva estructura de proyecto de C# dotnet core para que se complete con un archivo `Program.cs`, un archivo `workspace.csproj` y la carpeta `obj`.
21. Abra el archivo `Program.cs` y reemplace el contenido por el código siguiente.

C#Copiar

```
//
```

```
// Copyright (c) Microsoft. All rights reserved.
```

```
// Licensed under the MIT license. See LICENSE.md file in the project  
root for full license information.
```

```
//  
  
using System;  
  
using System.Threading.Tasks;  
  
using Microsoft.CognitiveServices.Speech;  
  
using Microsoft.CognitiveServices.Speech.Audio;  
  
namespace HelloWorld  
{  
    class Program  
    {  
        // It's always a good idea to access services in an async fashion  
  
        static async Task Main()  
        {  
            await RecognizeSpeechAsync();  
        }  
  
        static async Task RecognizeSpeechAsync()
```

```
{

    // Configure the subscription information for the service to
    access.

    // Use either key1 or key2 from the Speech Service resource you
    have created

    var config =
    SpeechConfig.FromSubscription("YourSubscriptionKey",
    "YourServiceRegion");

    // Setup the audio configuration, in this case, using a file that is
    in local storage.

    using (var audioInput =
    AudioConfig.FromWavFileInput("YourAudioFileName"))

    // Pass the required parameters to the Speech Service which
    includes the configuration information

    // and the audio file name that you will use as input

    using (var recognizer = new SpeechRecognizer(config,
    audioInput))

    {

        Console.WriteLine("Recognizing first result...");
```



```
var result = await recognizer.RecognizeOnceAsync();

switch (result.Reason)
{
    case ResultReason.RecognizedSpeech:

        // The file contained speech that was recognized and
the transcription will be output

        // to the terminal window

        Console.WriteLine($"We recognized: {result.Text}");

        break;

    case ResultReason.NoMatch:

        // No recognizable speech found in the audio file that
was supplied.

        // Out an informative message

        Console.WriteLine($"NOMATCH: Speech could not be
recognized.");

        break;

    case ResultReason.Canceled:

        // Operation was cancelled
```

```
// Output the reason

var cancellation =
CancellationDetails.FromResult(result);

Console.WriteLine($"CANCELED:
Reason={cancellation.Reason}");

if (cancellation.Reason == CancellationReason.Error)
{
    Console.WriteLine($"CANCELED:
ErrorCode={cancellation.ErrorCode}");

    Console.WriteLine($"CANCELED:
ErrorDetails={cancellation.ErrorDetails}");

    Console.WriteLine($"CANCELED: Did you update the
subscription info?");
}

break;
}

}

}

}
```

}

22. En la ventana de terminal, escriba el comando siguiente `dotnet add package Microsoft.CognitiveServices.Speech` y presione Entrar.
23. En la ventana de terminal, escriba el siguiente comando para agregar también el paquete de audio, `dotnet add package Microsoft.CognitiveServices.Speech.Audio`, y presione Entrar.
24. Visual Studio Code instala el paquete para su uso con la clase `AudioConfig` que se usará en el código.
25. Escriba la instrucción `using` siguiente para incluir el paquete en `Program.cs`.

19

C#Copiar

```
using Microsoft.CognitiveServices.Speech.Audio;
```

26. En el archivo `Program.cs`, pegue la clave del servicio de Voz en la que el texto **YourSubscriptionKey** es lo siguiente:
27. Escriba la ubicación como, por ejemplo, **westus** en la opción **YourServiceRegion** de esa misma línea.
28. Busque el texto **YourAudioFileName** y reemplácelo por **narration.wav**.

29. En la ventana de terminal, escriba `dotnet run` y, si se ha escrito correctamente, debería ver el resultado de la traducción en la ventana de terminal.

VISTA DE IDIOMAS DE VOZ ADMITIDOS

Los servicios de traducción de voz están diseñados para escenarios de traducción de idioma específicos del destino. Los idiomas de voz admitidos se describen mediante los siguientes tipos: *conversión de voz en texto*, *traducción de voz y texto a voz*. La compatibilidad con el idioma varía según la capacidad del servicio de Voz.

Por ejemplo, en conversión de voz en texto, encontrará compatibilidad con el inglés en seis configuraciones regionales:

- Inglés (Australia): en-AU
- Inglés (Canadá): en-CA
- Inglés (Reino Unido): en-GB
- Inglés (India): en-IN
- Inglés (Nueva Zelanda): en-NZ
- Inglés (Estados Unidos): en-US

A la hora de decidir cuáles son los servicios que usará para sus necesidades, visite la [página de idiomas y regiones admitidos](https://dev.microsofttranslator.com/languages?scope=speech) para obtener información actualizada sobre los idiomas admitidos. También se puede detectar mediante programación la compatibilidad con idiomas mediante una cadena de consulta, <https://dev.microsofttranslator.com/languages?scope=speech>

h,text,tts&api-version=3.0, que devuelve los idiomas admitidos en formato JSON que se pueden analizar. Aquí se muestra un fragmento de código de los resultados devueltos:

JSONCopiar

```
{  
  
  "translation": {  
  
    "af": {  
  
      "name": "Afrikaans",  
  
      "nativeName": "Afrikaans",  
  
      "dir": "ltr"  
  
    },  
  
    "ar": {  
  
      "name": "Arabic",  
  
      "nativeName": "العربية",  
  
      "dir": "rtl"  
  
    },  
  
    "bg": {  
  
      "name": "Bulgarian",  
  
      "nativeName": "Български",
```

```
    "dir": "ltr"

  },

  "bn": {

    "name": "Bangla",

    "nativeName": "বাংলা",

    "dir": "ltr"

  },

  "bs": {

    "name": "Bosnian",

    "nativeName": "bosanski (latinica)",

    "dir": "ltr"

  },

  "ca": {

    "name": "Catalan",

    "nativeName": "Català",

    "dir": "ltr"

  }

}
```

}

En la página de idiomas admitidos se muestran los idiomas admitidos por el SDK de Voz y la API REST, por lo que puede tener compatibilidad con su método preferido de acceso a los servicios. También puede realizar alguna personalización, para un subconjunto de idiomas, a fin de ayudar a mejorar la precisión. Se ofrece la personalización para un subconjunto de los idiomas a través de la carga de Transcripciones etiquetadas de audio y voz humana o Texto relacionado: Oraciones. La personalización de la pronunciación solo está disponible actualmente para en-US y de-DE. Obtenga más información sobre la personalización [aquí](#).

EJERCICIO OPCIONAL: ESCUCHA DE DATOS DE TRADUCCIÓN ENTRANTES

Otro uso potencial de Conversión de voz en texto es realizar la operación de traducción en la voz entrante que se escucha desde un micrófono. Este escenario puede producirse en una sala de presentaciones en la que el servicio de Voz proporciona la traducción de voz a texto de las personas que hablan y, después, lo muestra como subtítulos en una pantalla de proyección. En este ejercicio, verá un ejemplo de la escucha de voz desde un micrófono y su traducción a la salida de texto.

Nota

Este es un ejercicio opcional, ya que requiere el uso de un equipo local y la instalación de software para que los pasos del ejercicio se realicen correctamente.

Para probar este escenario, el entorno en línea creará incidencias con el acceso al micrófono, por lo que en este caso usaremos el equipo local. Como consecuencia, deberá tener instalado Visual Studio o Visual Studio Code. Dado que Visual Studio Code es una herramienta de desarrollo gratuita compatible con Windows, Mac y Linux, será la herramienta que usaremos. Visite la página de [Visual Studio Code](#) para descargar e instalar la aplicación en el equipo local. Una vez que se haya finalizado la instalación, inicie el ejercicio.

1. Para este ejercicio, cree una carpeta local en la que se va a almacenar el proyecto.
2. Inicie Visual Studio Code y abra la carpeta que se ha creado en el paso 1.
3. Abra una ventana de terminal en VS Code mediante la combinación de teclas CTRL+` (Control más la tilde aguda). Opcionalmente, puede optar por el menú Ver y, después, Terminal.
4. Escriba el comando `dotnet new console` y presione Entrar.
5. Dotnet Core crea una estructura de proyecto en su carpeta y agrega un archivo `Program.cs`.
6. En la ventana de terminal, escriba el comando siguiente, `dotnet add package Microsoft.CognitiveServices.Speech` y presione Entrar.
7. El paquete necesario se agregará al proyecto y se requerirá para tener acceso al SDK de Voz.
8. Pegue el código siguiente en `Program.cs` y reemplace el código existente.

C#Copiar

//

// Copyright (c) Microsoft. All rights reserved.

```
// Licensed under the MIT license. See LICENSE.md file in the project  
root for full license information.
```

```
//
```

```
using System;
```

```
using System.Threading.Tasks;
```

```
using Microsoft.CognitiveServices.Speech;
```

```
namespace helloworld
```

```
{
```

```
    class Program
```

```
    {
```

```
        public static async Task RecognizeSpeechAsync()
```

```
        {
```

```
            var config =  
SpeechConfig.FromSubscription("YourSubscriptionKey",  
"YourServiceRegion");
```

```
            using (var recognizer = new SpeechRecognizer(config))
```

```
{  
  
    var result = await recognizer.RecognizeOnceAsync();  
  
    if (result.Reason == ResultReason.RecognizedSpeech)  
  
    {  
  
        Console.WriteLine($"We recognized: {result.Text}");  
  
    }  
  
    else if (result.Reason == ResultReason.NoMatch)  
  
    {  
  
        Console.WriteLine($"NOMATCH: Speech could not be  
recognized.");  
  
    }  
  
    else if (result.Reason == ResultReason.Canceled)  
  
    {  
  
        var cancellation = CancellationDetails.FromResult(result);  
  
        Console.WriteLine($"CANCELED:  
Reason={cancellation.Reason}");  
  
  
  
        if (cancellation.Reason == CancellationReason.Error)
```

```

        {

            Console.WriteLine($"CANCELED:
ErrorCode={cancellation.ErrorCode}");

            Console.WriteLine($"CANCELED:
ErrorDetails={cancellation.ErrorDetails}");

            Console.WriteLine($"CANCELED: Did you update the
subscription info?");

        }

    }

}
}

```

```

static void Main()

{

    Console.WriteLine("Begin speaking....");

    RecognizeSpeechAsync().Wait();

    Console.WriteLine("Please press <Return> to continue.");

    Console.ReadLine();

}

```

```
}
```

```
}
```

9. Reemplace **"YourSubscriptionKey"** por la clave del servicio de Voz que se ha creado en este módulo.
10. Reemplace **"YourServiceRegion"** por la región en la que se ha creado el servicio de Voz, como *westus*.
11. Asegúrese de que tiene un micrófono conectado al equipo local y que funciona.
12. En el terminal, escriba `dotnet run` para iniciar la aplicación.
13. Empiece a hablar para que la aplicación pueda recopilar el audio de streaming.
14. Cuando termine, debería ver la salida de texto transcrito en el terminal.
15. Cuando se le solicite, presione Entrar para salir de la aplicación.