

O'REILLY®

# Guía del desarrollador para crear aplicaciones de IA

**Cree su primer bot inteligente  
con la IA de Microsoft**



**Anand Raman  
y Wee Hyong Tok**



# LA PLATAFORMA DE IA DE MICROSOFT

Cree aplicaciones inteligentes

Productividad de  
inteligencia artificial  
para todos los  
desarrolladores  
y cualquier  
escenario



Más información sobre la IA de Microsoft

**[azure.com/ai](https://azure.com/ai)**

Descubra cómo crear inteligencia en sus soluciones  
usando la plataforma de IA de Microsoft

**[aischool.microsoft.com](https://aischool.microsoft.com)**

---

# Guía del desarrollador para aplicaciones de IA

*Cree su primer bot inteligente  
con la IA de Microsoft*

*Anand Raman y Wee Hyong Tok*

Pekín • Boston • Farnham • Sebastopol • Tokio

**O'REILLY®**

---

# Contenido

<b>Guía del desarrollador para aplicaciones de IA .....</b>	<b>1</b>
Introducción.....	1
La intersección del cloud, los datos y la IA.....	5
La plataforma de IA de Microsoft.....	10
Cómo desarrollar un chatbot inteligente.....	12
Cómo añadir la inteligencia plug and play a su bot.....	36
Cómo crear una aplicación comercial para obtener conocimientos del bot: el panel de Conference Buddy.....	39
Cómo allanar el camino .....	46



---

# Guía del desarrollador para aplicaciones de IA

## Introducción

La inteligencia artificial se está convirtiendo rápidamente en una tecnología puntera que está ayudando a transformar el panorama y a avanzar. Viajemos al remoto Nepal para conocer un ejemplo fascinante. Al igual que la gran mayoría de nepaleses, Melisha Ghimere proviene de un pueblo remoto de una familia de granjeros que se dedicaba a la cría de reses, vacas y búfalos acuáticos. Hace siete años vio cómo sus acomodados tíos perdieron muchos de sus rebaños a causa de un foco de carbunco bacteridiano y su economía nunca se recuperó. Melisha empezó a estudiar en la universidad preocupada por la difícil situación de su familia. En la universidad ayudó a desarrollar una solución predictiva de alerta temprana para ayudar a los granjeros. Junto con un equipo de cuatro estudiantes, estudiaron la actividad ganadera, las prácticas veterinarias y se pusieron en contacto con muchos granjeros. Crearon un prototipo de dispositivo de supervisión que hace un seguimiento de la temperatura, los patrones de sueño, los niveles de estrés, el movimiento y la actividad de los animales de granja. El sistema de IA de Melisha predice el estado de salud de cada animal basándose en cambios sutiles frecuentes en sus observaciones. Los granjeros pueden monitorizar a sus animales, recibir alertas y recomendaciones que pueden ponerse en práctica. Aunque su proyecto se encuentra en la fase de pilotaje, las pruebas de campo han demostrado que esta solución tenía una precisión de alrededor del 95% a la hora de predecir riesgos en la salud del animal. Melisha y su equipo pudieron ayudar a una familia a evitar un foco de carbunco bacteridiano mortal gracias a la identificación de una vaca enferma, antes de que apareciesen los primeros síntomas evidentes. El equipo de Melisha fue finalista regional en la competición Imagine Cup de Microsoft en 2016.<sup>1</sup>

---

<sup>1</sup> [The Future Computed—Artificial Intelligence and its role in society—Microsoft](#)

Ahora explicaré otro ejemplo más cercano de cómo el poder de la IA puede ayudar a transformar la experiencia minorista. Lowes Innovation Labs ha creado una experiencia de compra única para la remodelación de viviendas. Por ejemplo, un cliente ahora puede entrar y compartir sus fotos de la cocina de sus sueños con un diseñador. Utilizando una aplicación con IA, el diseñador puede conocer las preferencias y el estilo de cada cliente. La aplicación genera una coincidencia a partir de los deseos de Lowe y el diseño de la cocina aparece en una imagen holográfica muy realista basada en realidad virtual con unas gafas Hololens.<sup>2</sup> El cliente puede visualizar, explorar y cambiar el diseño según sus preferencias en el entorno de realidad virtual y en tiempo real. Aplicaciones como estas están a la vanguardia de la revolución en experiencias minoristas que la IA puede aportar a los clientes.

La asistencia sanitaria es otro campo que está a las puertas de la revolución. Gracias al poder de la IA y las diversas fuentes de datos procedentes de la genómica, los registros médicos electrónicos, la documentación médica y los datos de la población, los científicos ahora pueden predecir emergencias sanitarias, realizar mejores diagnósticos y optimizar los cuidados. Un ejemplo en esta área sería el de **Cochrane**, un organismo sin ánimo de lucro de muy buena reputación que se dedica a la recopilación y la sintetización de las mejores evidencias del campo de la investigación para ayudar a los médicos a tomar decisiones informadas sobre las opciones de tratamiento. Cochrane realiza revisiones sistemáticas, que ingieren y analizan documentación médica de crecimiento masivo, y la reduce a segmentos de trabajos breves y gestionables para ofrecer a los facultativos las mejores orientaciones posibles acerca de los efectos de las intervenciones de atención médica. Por ejemplo, una revisión sistemática reciente de estudios médicos analizaba si los esteroides podían ayudar a la maduración de los pulmones en bebés prematuros. Los resultados de la revisión fueron concluyentes: los esteroides podían ayudar a salvar la vida de estos bebés. Esta intervención ha ayudado a cientos de miles de bebés prematuros. Sin embargo, las revisiones son muy laboriosas y pueden tardar de dos a tres años en completarse. La iniciativa **Project Transform** de Cochrane nació para dar respuesta a la necesidad de hacer que las revisiones sistemáticas fuesen más eficientes, ofreciesen instrucciones más puntuales y relevantes a los médicos y, por lo tanto, ayudasen a salvar más vidas. Project Transform emplea la IA para manipular y analizar la documentación y los datos de forma muy eficaz y, por lo tanto, permite a los investigadores comprender los datos e interpretar los resultados. Crea una asociación perfecta entre la persona humana y la máquina, en la que se reduce una enorme cantidad de sobrecarga de revisiones sistemáticas, y las capacidades de análisis humano pueden dirigirse a donde más se necesitan para lograr niveles idóneos de puntualidad y calidad.

---

<sup>2</sup> <http://www.lowesinnovationlabs.com/hololens>

No existe ningún campo que no se verá afectado por el poder transformacional de la IA. Podemos citar campos tan diversos como la astronomía, donde la IA ha acelerado el ritmo de los nuevos descubrimientos, y el campo de la conservación, donde los ecologistas y los conservacionistas están trabajando con herramientas con IA para ayudar a controlar, estudiar y proteger a los animales amenazados y las especies en peligro.

**NOTA**

Muchas veces nos atascamos en discusiones sobre las herramientas o los algoritmos más adecuados, pero el poder real de la IA reside en las ideas y las preguntas que puede predecir.

Se trata de la consideración ecologista de cómo crear hábitats sostenibles, la decisión del médico sobre cómo tratar mejor a sus pacientes, la curiosidad como ciudadanos científicos que amplía nuestra concienciación colectiva a los límites externos del universo. LA IA tiene el potencial de capacitar las causas humanas más nobles y tan solo es el principio. Este campo todavía es emergente y sus avances resaltan el increíble poder de la IA a la hora de remodelar nuestras experiencias diarias, cómo hacemos negocios y cómo vivimos nuestras vidas.

Hace cinco décadas, los primeros inventores de la IA solo podían soñar con lo que los consumidores actuales dan por hecho. Desde asistentes de voz como Cortana, Siri o Alexa, a smartphones o vehículos sin conductor, parece que estamos viviendo en una película de ciencia ficción. ¿Qué nos deparan las dos siguientes décadas? ¿Y las cinco siguientes? En Microsoft, nos hemos fijado el objetivo de desarrollar innovaciones de IA democratizando las herramientas de IA de la misma forma que democratizamos el poder de la computación en la era de las computadoras: concibiendo un ordenador personal en cada hogar, escuela y lugar de trabajo.

Como educador y pionero en computación, Alan Kay dijo “La mejor forma de predecir el futuro es crearlo”. Con el mismo espíritu, escribimos este libro para ofrecer a los desarrolladores un primer paso hacia la creación del futuro con la IA. En este libro, mostraremos cómo crear la primera aplicación de IA en el cloud y durante el proceso conocer todas las funcionalidades y todos los recursos de IA cada vez más disponibles para los programadores. La aplicación que creemos será un bot alimentado por IA, un “compañero de reunión” que ayude a crear una nueva experiencia de preguntas y respuestas para los asistentes y los oradores que participan en una conferencia. A medida que creemos este bot, podemos hacernos una idea de cómo la IA puede ayudar a entender las conversaciones, a percibir grandes cantidades de información y a responder de forma inteligente. En el proceso, también conoceremos el panorama de las herramientas de IA y los nuevos avances en el campo.



Hemos escogido a un chatbot como nuestro ejemplo porque es un punto de entrada relativamente fácil a la IA y en el proceso señalamos los recursos y los enlaces para ayudar a investigar más a fondo. Los chatbots también son omnipresentes con implementaciones interesantes y están transformando la forma en que interactuamos con los ordenadores. También se ofrece un panorama más amplio de las herramientas de IA y un vistazo a los nuevos avances en el campo.

A continuación ofrecemos un resumen de contenidos de este libro:

### *“La intersección del cloud, los datos y la IA”*

En el resto de esta sección, explicaremos la IA y la potente intersección de los datos, el cloud y las herramientas de IA que está creando un cambio de paradigma, lo que ayuda a crear sistemas de inteligencia.

### *“La plataforma de IA de Microsoft”*

En esta publicación exploraremos la plataforma de IA de Microsoft y señalaremos las herramientas, la infraestructura y los servicios que están disponibles para desarrollar aplicaciones de IA.

### *“Cómo desarrollar un chatbot inteligente”*

En esta sección se explican qué son los chatbots, la IA conversacional y algunas implementaciones de chatbots. ¿Cómo crear un chatbot inteligente para la empresa? Ofrecemos una arquitectura de alto nivel usando el ejemplo de bot Conference Buddy con muestras de código; debatimos las consideraciones de diseño y las tecnologías implicadas; y analizamos en profundidad la capa de abstracción del bot, lo que denominamos el Cerebro del bot.

### *“Cómo añadir la inteligencia plug and play a su bot”*

En esta sección se explica cómo conceder nuevas habilidades y funcionalidades al bot, como visión, traducción, discurso y otras capacidades de IA, así como cómo desarrollar la inteligencia del Cerebro del bot.

### *“Cómo crear una aplicación comercial para obtener conocimientos del bot: el panel de Conference Buddy”*

En esta sección se explica el funcionamiento del panel Conference Buddy, que permite a los oradores y los asistentes de una conferencia ver las preguntas de los asistentes y responderlas en tiempo real. También debatiremos cómo instrumentar al bot para obtener métricas y conocimientos sobre aplicaciones.

### *“Cómo allanar el camino”*

En la última sección trataremos un emocionante avance en el mundo de la IA con el lanzamiento de *Open Neural Network Exchange* (ONNX) y el compromiso de Microsoft con los seis principios éticos: equidad, fiabilidad y seguridad, privacidad y protección, inclusividad, transparencia y responsabilidad, para guiar el desarrollo interdisciplinar y el uso de la IA.

# La intersección del cloud, los datos y la IA

Definimos la IA como un conjunto de tecnologías que permiten que los ordenadores ayuden y solucionen problemas de formas similares a los humanos percibiendo, aprendiendo y razonando. Estamos haciendo posible que los ordenadores aprendan de grandes cantidades de datos y que interactúen de una forma más natural y receptiva con el mundo, en lugar de seguir las rutinas preprogramadas.<sup>3</sup> Se están desarrollando tecnologías para enseñar a los ordenadores a ver, escuchar, entender y razonar.<sup>4</sup> Los principales grupos de funcionalidades incluyen:

## *Visión por ordenador*

Es la capacidad que tienen los ordenadores de “ver” reconociendo objetos y sus relaciones en una imagen o un vídeo.

## *Reconocimiento de voz y síntesis*

Es la capacidad que tienen los ordenadores de “escuchar” y entender las palabras que dicen las personas para poder transcribirlas en forma de texto, así como leer el texto en voz alta con una voz natural.

## *Comprensión del lenguaje*

La capacidad de los ordenadores para “comprender” el significado de las palabras y responder, teniendo en cuenta los matices y las complejidades del lenguaje (tales como jergas y expresiones idiomáticas). Cuando los ordenadores pueden participar de forma efectiva en un diálogo con seres humanos, lo llamamos “IA conversacional”.

## *Conocimiento*

La capacidad de un ordenador de comprender el “motivo” representando y entiendo la relación entre personas, cosas, lugares y eventos.

¿Qué representan estas funcionalidades en el contexto de las aplicaciones comerciales? El poder de la IA está impulsando las aplicaciones que *questionan los hechos* que liberan el poder de todos los datos recopilados en el tiempo, en repositorios y *datasets* masivos a través del *machine learning*. Estos sistemas potenciados por IA *entienden* y crean significado a partir de datos no estructurados tales como emails, chats y notas escritas a mano; cosa que no podía hacerse anteriormente. Y, lo que es más importante, los sistemas *están interactuando* con los clientes y los *capta* en varios canales y de distintas formas hiperpersonalizadas.

---

<sup>3</sup> Revista *Time*: Why you shouldn't be afraid of AI

<sup>4</sup> The Future Computed—Artificial Intelligence and its role in society—Microsoft

Paralelamente, los negocios están usando aplicaciones habilitadas con IA para transformar digitalmente todos los aspectos de sus organizaciones, como: transformando sus productos a través de conocimientos de datos de clientes, optimizando las operaciones comerciales prediciendo anomalías y mejorando la eficacia, capacitando a sus empleados a través de herramientas inteligentes, y captando a sus clientes a través de agentes de conversación que ofrecen experiencias más personalizadas.

Los siguientes ejemplos son preguntas que capacitan a los motores que ejecutan aplicaciones con IA:

#### *Clasificaciones*

¿A qué categoría pertenece?

#### *Regresión*

¿Cuánto? ¿Cuántos?

#### *Anomalía*

¿Es raro?

#### *Agrupación en clústeres*

¿Cómo se organiza?

Entonces, ¿cómo empezaría a diseñar soluciones con IA para aprovechar todas las funcionalidades mencionadas anteriormente? Diseñamos soluciones de IA para complementar y liberar el potencial humano y actividades creativas. Existen muchas implicaciones en el diseño de tecnologías para humanos y para ello tenemos que considerar las implicaciones éticas, comprender el contexto de la forma de trabajar, jugar y vivir de las personas, así como crear soluciones personalizadas que puedan adaptarse con el paso del tiempo.

Una de las áreas más fascinantes de la investigación es sincronizar la inteligencia emocional y cognitiva para poder crear sistemas de IA de conversión que modelen el lenguaje humano y cuenten con conocimientos sobre la interacción lógica e impredecible de las personas humanas.

En palabras de Lili Cheng, vicepresidenta corporativa de *Microsoft AI and Research*, “Esto supone que la IA tiene que poder reconocer cuándo las personas son más eficientes por su cuenta, cuándo debe apartarse y cuándo no es necesario ayudar, cuándo no es necesario grabar, cuándo no debe interrumpir o distraer”.<sup>5</sup>

Ha llegado la era de la IA y esta afirmación queda clara si tenemos en cuenta la proliferación de los datos, la disponibilidad limitada de los poderes de computación en el cloud y el aumento de los potentes algoritmos que están definiendo el futuro.

---

<sup>5</sup> Revista *Time*: Why you shouldn't be afraid of AI

# La IA moderna: intersección de datos, computación en el cloud e IA

Aunque la investigación de la IA se remonta varias décadas atrás, en los últimos años se ha producido un aumento de las innovaciones prácticas, catalizado por la enorme cantidad de datos digitales, servicios online y el enorme poder de la computación. Como resultado, las tecnologías como la comprensión del lenguaje natural, el análisis de sentimientos, el reconocimiento del discurso, la comprensión de imágenes y el *machine learning* ahora son lo suficientemente precisas para habilitar numerosas aplicaciones en un amplio abanico de industrias.

Pasemos a examinar tres importantes avances que están ayudando a crear la IA moderna: los datos y la transformación digital, la computación en el cloud, las herramientas y los algoritmos de IA.

## Los datos y la transformación digital

En el centro de la IA están los datos y el aumento de la digitalización de nuestra era está fomentando la proliferación de lo que conocemos como *big data*. De las cerca de 7,4 mil millones de personas que habitan la tierra, más de 4 mil millones tienen dispositivos móviles y 3,8 mil millones están conectadas a Internet, y se prevé que estas cifras sigan aumentando. La gran mayoría de nueva información en el mundo se genera y consume online, y una creciente cuota de la economía está migrando a los servicios online, de las compras a la banca, el ocio, los medios y las comunicaciones. Dado que nuestras vidas cada vez están más digitalizadas y los sensores (micrófonos, cámaras, localización y otros sensores) cada vez son más económicos y predominantes, ahora tenemos más datos que nunca para que los ordenadores puedan aprender y razonar. Paralelamente, a medida que participamos en transacciones e interacciones online digitalmente, generamos nuevos datos de respuesta y retroalimentación que permiten adaptar los algoritmos de IA y optimizar las interacciones. Paralelamente, a medida que participamos en transacciones e interacciones online digitales, generamos nuevos datos de respuesta y retroalimentación que permiten adaptar los algoritmos de IA y optimizar las interacciones.

La gran cantidad y el índice de crecimiento de los datos ha permitido conseguir grandes innovaciones en la forma de almacenar, gestionar y minar los datos eficientemente para realizar análisis flexibles en tiempo real. La mayoría de esos datos fluyen a través de *clouds* privadas o públicas por Internet. Los sistemas de “big data” ayudan a gestionar la naturaleza heterogénea de dichos datos y a habilitar muchos métodos de análisis, tales como el análisis estadístico, el *machine learning*, la minería de datos y el *deep learning*.

Estos sistemas están en el centro de lo que permite que los ordenadores “vean”, “escuchen” y “razonen”, así como para discernir patrones que a menudo son indetectables para el ojo humano.

## Informática en nube

Internet y la transformación digital del mundo han ayudado a catalizar la computación en el *cloud*. El procesamiento de los datos y el suministro de servicios online a gran escala requieren un poder de computación enorme, redes fiables, almacenamiento y procesamiento de datos. El *cloud* ofrece una base muy potente y una plataforma sólida para gestionar estos desafíos. Le permite racionalizar datos de dispositivos conectados, ofrece una capacidad de almacenamiento de datos masiva y una potencia de computación elástica y escalable para integrar, analizar y aprender de los datos.

También puede utilizar los servidores más grandes, las últimas GPU y el hardware más avanzado como *Field Programmable Gate Arrays* (FGPAs) para acelerar los cálculos más complejos sin el coste exorbitante de crear y aprovisionar centros de datos y granjas de servidores. Su enorme conectividad permite usar cualquier tipo de dispositivo (lo que conocemos como el Internet de las cosas (IoT)) para migrar enormes cantidades de datos al *cloud* en tiempo real para análisis e IA a escala. Además, el *cloud* ofrece la infraestructura y las herramientas necesarias para ofrecer seguridad, disponibilidad, conformidad y gestionabilidad de grado empresarial para las aplicaciones y los servicios desplegados en el *cloud*.

## Herramientas y algoritmos de IA

La explosión de casos de uso para IA accionada por servicios online y la transformación digital han catalizado el enorme progreso conseguido en los algoritmos de IA. Una de las innovaciones más profundas en los últimos años ha sido el *deep learning*. Esta técnica, inspirada en las redes neuronales del cerebro, permite que los ordenadores aprendan conceptos profundos, relaciones y afirmaciones de grandes cantidades de datos (tales como imágenes, vídeo y texto), así como a realizar tareas como reconocimiento de voz y objetos con una precisión comparable a la de los humanos. Actualmente, las herramientas de código abierto tales como Cognitive Toolkit, PyTorch y Tensorflow hacen que las innovaciones de *deep learning* sean accesibles para una amplia audiencia. Además, los principales proveedores del *cloud* ahora cuentan con servicios que simplifican considerablemente el desarrollo de IA para poder capacitar a los ingenieros de software.

La IA moderna reside en la intersección de estas tres potentes tendencias: datos digitales de los cuales la IA puede aprender, servicios alojados en el *cloud* que permiten las interacciones alimentadas por IA y mejoras continuas en

algoritmos que hacen que las funcionalidades de IA sean más potentes, al tiempo que habilitan las nuevas aplicaciones y los casos prácticos.

## Sistemas de inteligencia: un cambio de paradigma

En un **esclarecedor artículo**, Jerry Chen, de Greylock Partners, explora la idea de *los sistemas de inteligencia*, que están alimentados por la IA y que tienen por objeto recombinar varios *datasets*, procesos y flujos de trabajo para crear un nuevo contexto. Un ejemplo sería una aplicación que combinase los análisis web con datos de cliente y datos sociales para poder predecir el comportamiento del usuario final, combinar datos y ofrecer contenidos de forma más puntual. Las historias del principio de este capítulo son ejemplos de cómo se crean estos sistemas para poder generar nuevos conocimientos y valores en varios sectores (agricultura, venta minorista, asistencia sanitaria, etc.).

El aumento de los *sistemas de inteligencia* es un ejemplo de cómo la combinación de datos, la computación en el *cloud* y la IA nos abre las puertas a un mundo de sistemas de inteligencia online muy dominantes. Tanto si se trata de servicios de transporte compartido, venta minorista online, redes sociales, medios de comunicación y ocio, banca, inversiones, transporte, manufactura, asistencia sanitaria o gobiernos, estos sistemas de inteligencia mediarán, gestionarán y optimizarán todas las interacciones y todos los intercambios. El alcance de este cambio de paradigma escapa a la comprensión de todos nosotros. Recordemos las palabras del famoso historiador de la ciencia, Thomas Kuhn:

Dirigidos por un nuevo paradigma, los científicos adoptan nuevos instrumentos y buscan en nuevos sitios. Y, lo que es más importante, durante las revoluciones los científicos ven nuevas cosas cuando buscan con instrumentos conocidos en lugares donde nunca antes habían buscado. Es como si su comunidad profesional se hubiera transportado de repente a otro planeta donde los objetos familiares se ven bajo una luz diferente y se combinan con otros objetos desconocidos. (Kuhn, 1962)<sup>6</sup>

De la misma forma, los ingenieros, empresarios y dirigentes comerciales, capacitados con sistemas de inteligencia, ven los problemas familiares bajo una luz distinta y descubren nuevas oportunidades económicas.

También resulta útil analizar la transformación de la tecnología de la información empresarial en las últimas décadas:

### *Revolución del servidor de cliente → Sistemas de registros*

La revolución de cliente-servidor permitió el uso de la tecnología de la información por primera vez para gestionar los negocios. Primeros sistemas

---

<sup>6</sup> In *The Structure of Scientific Revolutions* (1962, 2ª ed. 1970)

de registros de organizaciones: Sistemas *Customer Relationship Management* (CRM); sistemas *Human Capital Management* (HCM) para RR.HH.; y sistemas *Enterprise Resource Planning* (ERP) para activos clave y finanzas.

#### *Revolución de Internet → Sistema de participación*

El aumento del uso de Internet, los dispositivos móviles y el chat nos ha permitido crear sistemas de participación que enlazan con los sistemas de registros e interactúan directamente con los clientes y proveedores.

#### *Revolución de la IA → Sistema de inteligencia*

Actualmente están apareciendo sistemas de inteligencia que integran los datos en todos los sistemas de registro, se conectan directamente con los sistemas de participación y crean sistemas que comprenden y razonan con los datos. Estos sistemas pueden generar flujos de trabajo y procesos de gestión, optimizar operaciones y generar interacciones inteligentes con los clientes, empleados, proveedores y partes interesadas.

La industria de la TI ha pasado de descifrar las preguntas básicas del *big data* y la infraestructura de la IA a abrirnos las puertas al crecimiento del *machine learning* y las plataformas de IA, tales como la plataforma de IA de Microsoft, como una capa de aplicación. Este movimiento inspirará la aparición de nuevos modelos comerciales basados en *big data*/IA, de la misma forma que esos negocios “habilitados por web”, y transformará la experiencia del cliente y del negocio de una forma que por ahora solo podemos imaginar. En la siguiente sección, exploraremos la plataforma de IA de Microsoft y hablaremos de las herramientas y los recursos disponibles para los desarrolladores de IA.

## **La plataforma de IA de Microsoft**

La plataforma de IA de Microsoft pretende llevar la IA a todos los desarrolladores y capacitarlos para que puedan innovar y acelerar su crecimiento con diversos servicios, infraestructuras y herramientas. Desde IA preintegrada (que no necesita prácticamente formación) a IA personalizada, la plataforma de IA abierta de Microsoft permite que los desarrolladores utilicen diversas herramientas y entornos de *deep learning* y *machine learning*.



imagen 1-1. La plataforma de IA de Microsoft (imagen cedida por Microsoft)

La plataforma está formada por los siguientes servicios (ilustrados en la imagen 1-1):

### IA personalizada

Azure Machine Learning le permite desarrollar modelos de *machine learning* y *deep learning*, formarlos en el cloud y ponerlos en práctica. Azure dispone de muchos servicios de computación y datos para almacenar y procesar sus datos.

Azure Machine Learning también ofrece un servicio de experimentación que le permite empezar a crear prototipos rápidamente en el escritorio y escalarlos para ejecutar máquinas virtuales de *deep learning*, clústeres Spark y los servicios de IA de Azure Batch. Además, Azure Machine Learning le permite gestionar el rendimiento del modelo y el ciclo de vida, así como colaborar y compartir soluciones usando Git. Los contenedores *docker* le permiten desplegar modelos en producción más rápidamente en el cloud, in situ o en dispositivos periféricos inteligentes.

### IA preintegrada

Puede utilizar servicios “acabados” de alto nivel que aceleran el desarrollo de aplicaciones inteligentes, con tan solo unas pocas líneas de código. Estos servicios se personalizan en función de los requisitos de conformidad, seguridad y disponibilidad de cada organización. Cognitive Services proporciona IA preintegrada a través de una serie de API, SDK y servicios. Esto permite a los desarrolladores infundir la IA en las aplicaciones, los sitios web, los bots y mucho más.

### Bot Framework

Proporciona herramientas para acelerar el desarrollo para la IA conversacional. Se integra perfectamente con Cortana, Office 365, Slack, Facebook Messenger y otros.



Bot Framework es una plataforma para construir, conectar, probar y desplegar bots inteligentes y potentes. Gracias a su compatibilidad con .NET, Node.js y REST, podrá conseguir Bot Builder SDK y empezar a crear bots rápidamente con Bot Framework. Además, puede aprovechar los Microsoft Cognitive Services para añadir funciones inteligentes como la comprensión del lenguaje natural, el reconocimiento de imágenes, el discurso y mucho más.

Entre las herramientas de gestión y codificación de la plataforma de IA de Microsoft podemos encontrar Visual Studio Tools for AI, que le permite crear, depurar, probar y desplegar la IA con Visual Studio Code en Windows y Mac. Además, podrá beneficiarse de varias herramientas como Jupyter Notebooks, PyCharm y otras.

La plataforma de IA de Microsoft también integra varios entornos de *machine learning* y *deep learning*, como TensorFlow, Caffe2, Microsoft Cognitive toolkit (CNTK), Scikit-learn, MXNet, Keras y Chainer.

Para ayudarle a empezar, puede utilizar los recursos que están disponibles en la [Azure AI Gallery](#), que proporciona plantillas de soluciones *end-to-end*, esquemas arquitectónicos de referencia y códigos de muestra.

En la siguiente sección exploraremos cómo puede desarrollar su siguiente aplicación de inteligencia usando la plataforma de IA de Microsoft. Como ejemplo, desarrollaremos un chatbot inteligente y hablaremos de los chatbots/la IA conversacional, contaremos algunas historias de usuarios interesantes, de consideraciones de diseño y de cómo desarrollar la inteligencia del chatbot.

## Cómo desarrollar un chatbot inteligente

La reciente explosión y popularidad de los chatbots pone de relieve nuestra naturaleza esencial como seres humanos. En lugar de rellenar informes, seleccionar elementos en pantallas o navegar por páginas web poco intuitivas, las marcas han puesto a disposición numerosos servicios a través de una interfaz de cuadro de diálogo usando la voz o el texto. Piense en los bots como si fuesen aplicaciones con las que puede hablar. Los chatbots interactúan con usted fácilmente en una conversación y ayudan a responder sus preguntas o llevar a cabo tareas como confirmar sus reservas, pedir comida o comprar un artículo. Si se diseñan correctamente, pueden hacerle pensar que está hablando con una persona humana.

Algunos ejemplos de chatbots en el comercio minorista serían el bot de UPS, lanzado por el gigante de servicios de paquetería, que permite a los clientes obtener información que necesitan sobre sus paquetes, envíos, tarifas y oficinas de UPS. El bot de Macy proporciona a los clientes información sobre los pedidos, el merchandise, preguntas frecuentes e incluso realiza acciones como aplicar cupones y descuentos en la cesta de la compra. Dixon's Carphone, un gran proveedor de servicios y distribuidor minorista de telecomunicaciones y componentes

electrónicos en Europa, quería acortar distancias entre la investigación online de sus clientes y la experiencia en las tiendas. La empresa creó un chatbot denominado Cami para ayudar a racionalizar la experiencia del cliente dando a los empleados acceso a lo que buscaban online y dirigiéndoles a los productos o recomendándoles productos nuevos de la tienda. En el sector de la distribución minorista, vemos cómo los chatbots aumentan las capacidades de la plantilla existente, reduciendo los costes y el tiempo asociado al soporte técnico, y transformando la experiencia global del cliente.<sup>7</sup>

Existen nuevas aplicaciones, como las desarrolladas por la empresa japonesa de tecnología de navegación, NAVTIME Japan, que creó un chatbot asistente personal cuya misión era ayudar a los turistas a planificar sus viajes en tiempo real a medida que viajaban por Japón. El chatbot responde preguntas como “¿Dónde puedo ir a cenar?” o “¿Dónde puedo comprar souvenirs?”. Posteriormente, ofrece servicios de geolocalización para que el usuario sepa si está cerca de algún lugar que quería visitar. Visitar un país donde no se habla el idioma local ya no tiene por qué ser una experiencia desagradable.<sup>8</sup>

Las empresas utilizan la “conversación como plataforma”, es decir, bots que entienden el lenguaje humano o que utilizan el lenguaje como interfaz principal para facilitar las operaciones internas. Desde asistentes virtuales de RR.HH. a responder a preguntas rutinarias de los empleados y ayudar con las tareas de reclutamiento haciendo preguntas y enviando currículos, hasta bots de productividad como asistentes digitales y de reuniones. Los fabricantes utilizan los bots para conectarse con el IoT y ayudar a los empleados, de forma que éstos puedan vigilar las anomalías y recibir alertas de mantenimiento predictivo.

Después de más de diez años de estudiar los matices del lenguaje humano y las tecnologías que facilitan la interacción entre la IA y el ser humano, Microsoft ha creado Xiaoice: un chatbot chino muy popular con millones de seguidores. Parte de su popularidad se debe a la forma en que muestra su coeficiente emocional (EQ) recordando segmentos de conversaciones y haciendo un seguimiento en conversaciones posteriores. Los ciudadanos jóvenes en China utilizan todos los días Xiaoice para hablar de sus problemas, preocupaciones y presiones diarias y algunos de ellos han llegado decir que querían a su bot.<sup>9</sup> El talento y la popularidad de Xiaoice no conoce límites: ha publicado un libro de poesía, ha ayudado a los ciudadanos chinos a escribir más de 200 millones de poemas y actualmente presenta un noticiero matutino en televisión que cuenta con más de 800 millones de espectadores.

---

<sup>7</sup> Cómo crear bots y chatbots en el sector minorista

<sup>8</sup> Una empresa de tecnología de navegación utiliza chatbots para ayudar a los turistas que viajan por Japón

<sup>9</sup> Cada vez más ciudadanos chinos recurren a un programa de smartphone

Aunque los asistentes digitales como Cortana, Siri y Alexa han demostrado tener un alto coeficiente intelectual en sus funciones basadas en tareas, Xiaoice ilustra el potencial de combinar la inteligencia del coeficiente intelectual con la empatía del coeficiente emocional. Rinna se ha unido a la moda de Xiaoice en Japón. Cuenta con la misma popularidad y ahora domina los juegos de Pokemon de Nintendo. También tenemos a Zo en los Estados Unidos que anima a los usuarios a participar en batallas de rap.

Todavía queda mucho trabajo por hacer, nuevos estudios por realizar y nuevas oportunidades por descubrir para aumentar el coeficiente emocional en los sistemas informáticos. De esta forma, podremos traducir los matices del lenguaje humano con sus sutiles cambios de tono y variaciones de significado en contextos específicos.

La visión tan ambiciosa de Microsoft se extiende más allá de estos chatbots para crear una IA conversacional como plataforma, que pone el lenguaje natural en el centro de la computación. Estos sistemas se impregnarán con sentidos alimentados por la IA para crear una relación fluida entre el hombre y la máquina.

## **Evolución del procesamiento del lenguaje natural y los bots**

Demos un paso atrás y discutamos el procesamiento del lenguaje natural (NLP) y cómo, sin el inmenso avance en las áreas del NLP o comprensión del lenguaje natural (NLU), los bots tendrían muy pocas habilidades. El lenguaje humano suele ser desordenado, impreciso y vago sin referencias explícitas en la mayor parte del discurso. Hasta hace relativamente poco, el procesamiento y análisis del lenguaje natural ha sido muy complicado para los ordenadores, a pesar de los intentos continuados de los científicos informáticos desde 1950 de resolver el problema del procesamiento y análisis de componentes textuales, los sentimientos, las partes del discurso y las diversas entidades que componen el discurso. Los avances recientes en *machine learning* y la disponibilidad de grandes cantidades de texto digital y datos conversacionales a través de webchats, sistemas de mensajería y servicios como Twitter nos han ayudado a conseguir un gran avance en NLP.

NLP es básicamente la capacidad de coger un cuerpo de texto generado por un humano y transformarlo en lenguaje legible por máquina. NLP analiza y extrae los principales metadatos del texto, incluyendo lo siguiente:

### *Entidades*

NLP detecta nombres de personas, lugares y cosas.

### *Relaciones*

Identifica la relación entre las entidades.

### *Conceptos*

NLP extrae conceptos generales del cuerpo de texto que no aparecen explícitamente. Por ejemplo, la palabra “Excel” puede devolver conceptos

como "Herramientas de productividad" y "Números", incluso si estos términos no aparecen en el texto. Se trata de una herramienta muy potente para establecer conexiones que pueden no parecer obvias en un primer momento.

### *Sentimiento*

Hace referencia al nivel de positividad o negatividad en el texto. Resulta útil, por ejemplo, para evaluar el sentimiento acerca de un producto o servicio. O, en un contexto de atención al cliente, para evaluar cuándo es necesario pasar la conversación a una persona humana tras detectar un valor de negatividad.

### *Emociones*

Se trata de un análisis de sentimiento a nivel granular. Clasifica valores "positivos" y "negativos", además de ira, tristeza y alegría.

### *Palabras clave*

NLP extrae palabras clave y frases para usarlas como base de indexación, búsqueda, clasificación, etc.

### *Categorías*

Crea una taxonomía jerárquica para el asunto del contenido y lo coloca en una categoría de alto nivel (clasificación de texto). Resulta útil para aplicaciones como recomendar contenido relevante, generar anuncios, ordenar emails, etc.

Históricamente, podría implementar las funcionalidades de NLP programando las normas directamente, lo que hace que sean difíciles de adaptar a nuevos datos o escalas, o podría usar los modelos de *machine learning*. No obstante, la formación de modelos de *machine learning* requiere tener acceso a especializaciones poco frecuentes, *datasets* largos y herramientas complejas. Todo ello ha limitado su implementación a únicamente grandes empresas que pueden afrontar el gasto.

La disponibilidad de las funcionalidades de NLP como los extractores de entidades y los clasificadores de texto como APIS fáciles de usar en el *cloud* ha potenciado el aumento del uso de los chatbots. Desde la aparición de las herramientas de código abierto hasta la llegada de las API del *cloud*, las funcionalidades de NLP que anteriormente estaban limitadas a círculos académicos y la comunidad de investigación ahora está accesible para una audiencia más amplia en varios sectores.

Un ejemplo interesante de NLP en el *cloud* es el servicio **Language Understanding (LUIS)** desarrollado por Microsoft. LUIS utiliza el *machine learning* para permitir que los desarrolladores creen aplicaciones que puedan utilizar la información aportada por el usuario en lenguaje natural y extraer su significado y su intención. Una aplicación de cliente que conversa con el usuario

puede traducir la información aportada por el usuario en una aplicación LUIS y extraer los conceptos clave expresados por el usuario.

Al igual que ocurre con otros servicios cognitivos en la plataforma de IA de Microsoft, LUIS cubre todo el espectro de experiencias del desarrollador. Podrá utilizar un modelo preconfigurado, personalizar un modelo existente o crear su propio modelo desde cero. Un modelo empieza con una lista de intenciones generales del usuario como “reservar un vuelo” o “contactar con atención al cliente”. Después de identificar el objetivo, deberá proporcionar frases denominadas enunciados (que son la esencia del diálogo) para el propósito. Posteriormente, tendrá que etiquetar los enunciados con detalles específicos que usted desee que LUIS extraiga del enunciado. Con los dominios preconfigurados, tendrá un amplio conjunto de entidades y enunciados para categorías comunes como calendario, ocio, comunicación, automatización del hogar, etc.

Los sistemas como LUIS han sido diseñados para extraer los siguientes principios:

#### *Capacidad de reconocer intenciones*

¿Cuál es el objetivo del usuario? La intención es un objetivo o un propósito expresado como información del usuario, como reservar un vuelo, pagar una factura o encontrar un artículo de noticias.

Usted define y nombra intenciones que se corresponden con estas acciones. Una aplicación de viajes podría definir una intención denominada “Reserva de vuelos”.

#### *Capacidad de reconocer enunciados (diálogo)*

Los enunciados o el diálogo son datos introducidos por el usuario que su aplicación necesita para poder comprenderlos. Pueden ser una frase como “Reservar un billete a París” o un fragmento de una frase como “Reserva” o “Vuelo a París”. Los enunciados y los diálogos no siempre están bien formados y pueden existir muchas variaciones de diálogo para una intención.

#### *Capacidad de reconocer entidades*

Una entidad representa información detallada que es relevante para el diálogo. Por ejemplo, en el diálogo “Reservar un billete a París”, “París” es la entidad de lugar. LUIS reconoce y etiqueta las entidades que se mencionan en el diálogo del usuario para que usted pueda escoger la acción específica que desea adoptar para responder a la solicitud del usuario.

LUIS también permite a los desarrolladores mejorar de forma continua la aplicación a través del *active learning*. También integra otras herramientas de IA en el *cloud* para integrar el idioma natural en las aplicaciones, los bots y los dispositivos del IoT. Microsoft ofrece un conjunto adicional de herramientas a través de su *Bot Framework* para poder incorporar LUIS y otros servicios cognitivos en el desarrollo de bots. El *Bot Framework* de Microsoft ofrece un entorno de desarrollo integrado (IDE) que le permite crear, conectar, probar, desplegar y gestionar bots inteligentes desde un lugar.

Los clientes, preparados con sus experiencias con los asistentes digitales y el extendido uso de las aplicaciones de mensajería, cada vez interactúan más con los bots. Los bots proporcionan una experiencia del usuario mejorada porque pueden responder de una forma más rápida y efectiva a las solicitudes del usuario. Para muchas compañías, los bots están pasando a ser un diferenciador competitivo. Como hemos explicado anteriormente, muchas empresas están integrando estratégicamente los chatbots en sus plataformas de mensajería con los clientes para los descansos del personal. Analicemos un caso práctico de un bot: el Stack Overflow.

## Su primer bot: escenario

Ahora pasaremos a explicar cómo puede crear su primer bot. Imagine que está asistiendo a una presentación de una conferencia sobre tecnología con cientos de asistentes entusiasmados. A medida que el orador realiza la presentación, usted confecciona una lista de preguntas. Quiere hacer sus preguntas pero:

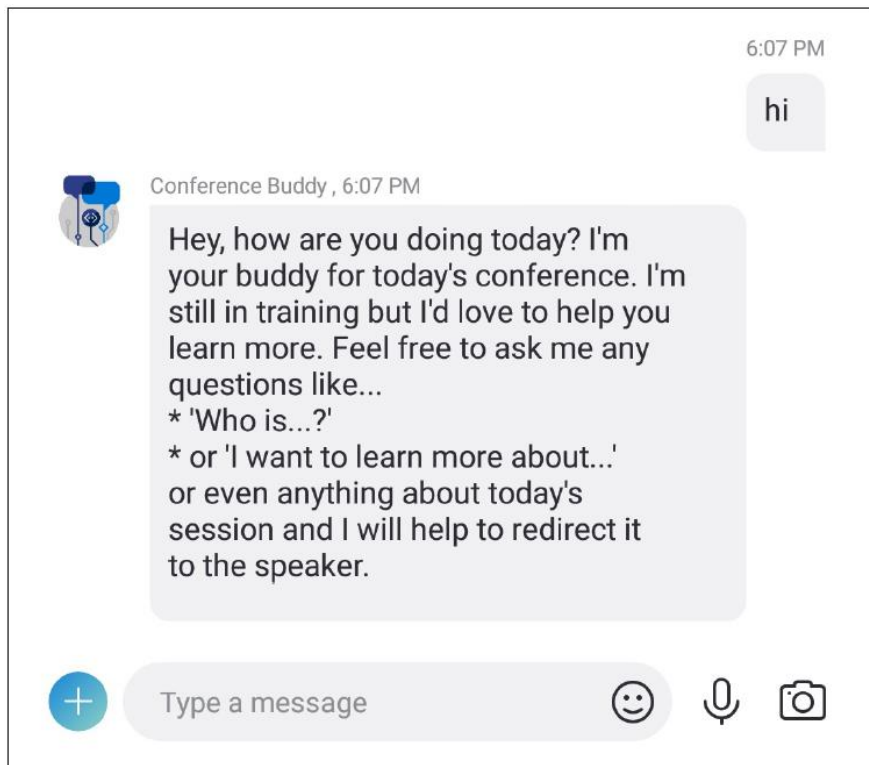
- No es el momento de ruegos y preguntas.
- No le entusiasma la idea de hablar en público.
- No levantó su mano lo suficientemente alto o sus dudas no fueron atendidas en la sesión de ruegos y preguntas.
- Existe una barrera idiomática y no puede comunicarse con fluidez en el idioma nativo del orador.

Los motivos son innumerables. La mayoría de personas no tienen la oportunidad de interactuar con el orador y no comprenden todos los contenidos durante la conferencia porque existen barreras logísticas y de otra índole.

¿Qué le parecería tener un chatbot Conference Buddy que pudiese hacer sus preguntas a medida que aparecen y obtener respuestas mientras usted sigue con sus tareas? Esas preguntas van a parar a un panel donde el orador puede verlas y responder a las dudas de la audiencia en tiempo real.

El cliente de chat Conference Buddy que vamos a crear tendrá tres funciones:

1. Responder sus dudas y presentarse él mismo, como se muestra en [Imagen 1-2](#).



*Imagen 1-2. Saludo del bot Conference Buddy*

2. Responde algunas de sus preguntas de forma automática e inteligente, siempre que es posible, como se demuestra en [Imagen 1-3](#).

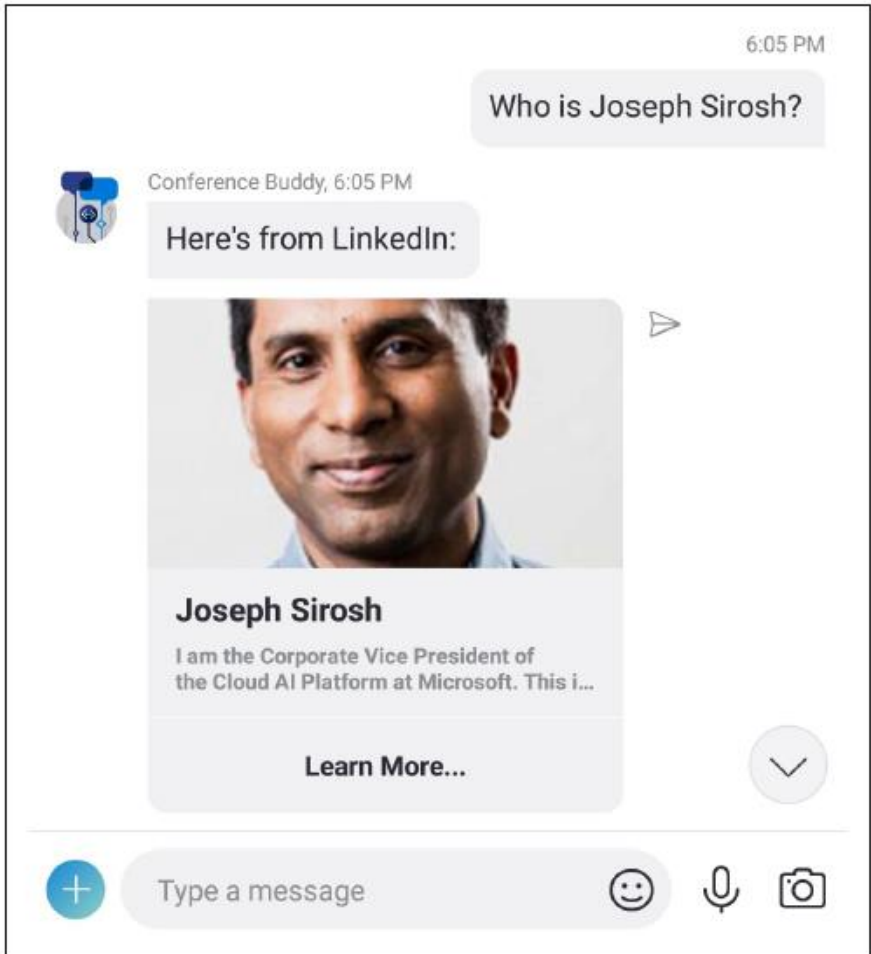


Imagen 1-3. Pregunta del bot Conference Buddy

3. Envíe sus preguntas al orador a través de un panel donde el orador verá todas las preguntas de la audiencia y escoger la pregunta que desea responder, como se muestra en [Imagen 1-4](#).



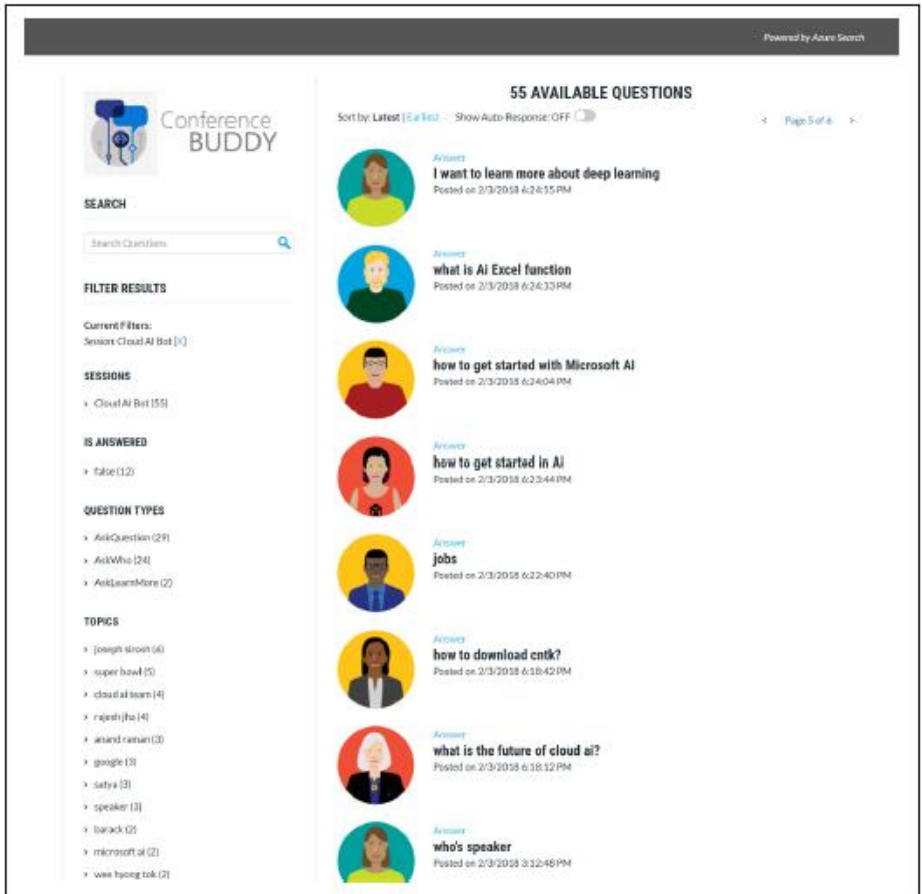


Imagen 1-4. La aplicación del panel Conference Buddy

Para hacerse una idea de cómo es el aspecto y el funcionamiento de esta aplicación, puede visitar la página web de Github <https://aka.ms/conferencebuddy> y verá una demostración y podrá revisar el código para esta muestra.

## Resumen de la arquitectura del bot Conference Buddy

Utilizaremos una arquitectura de microservicios para crear nuestro bot (Imagen 1-5) de forma que cada componente pueda crearse, implementarse y escalarse independientemente.

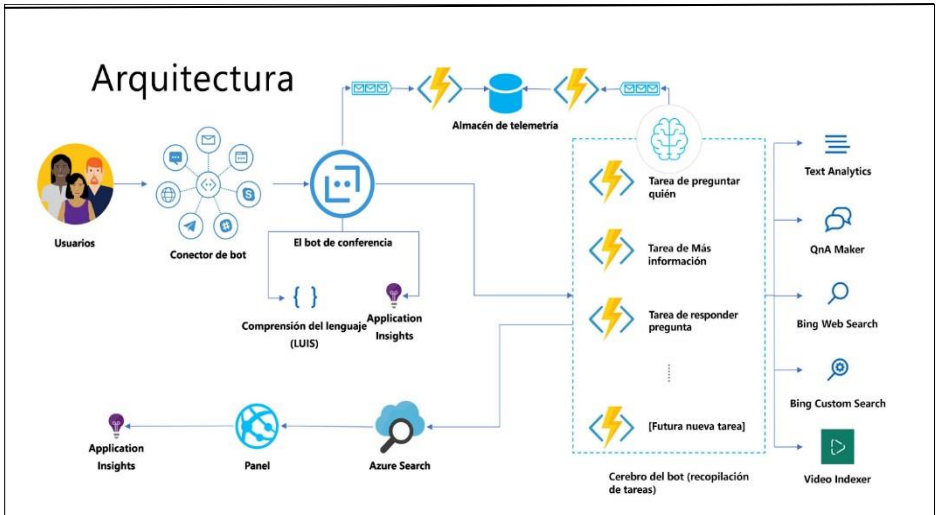


Imagen 1-5. Arquitectura del bot Conference Buddy

En nuestro bot Conference Buddy tenemos dos componentes principales:

### El bot de conferencia

Este componente gestiona todos los eventos relacionados con mensajes de forma inteligente.

### Cerebro del bot

Este componente abstrae la lógica comercial en el bot. Dentro del cerebro del bot encontramos las tareas individuales del bot.

Las preguntas y los resultados aparecen en el panel de Conference Buddy. Analicemos estos componentes individualmente.

### El bot de conferencia

El bot de conferencia, integrado en *Bot Framework*, gestiona todos los eventos de mensajes de participantes de forma inteligente. El bot tiene varios canales, lo que hace que los usuarios puedan enviar correos electrónicos, hablar por Skype o usar un servicio de mensajería personalizado que se conectará con el conector del bot para sincronizarse con el bot de conferencia. **Imagen 1-6** Muestra un saludo cuando se lanza la aplicación Conference Buddy.

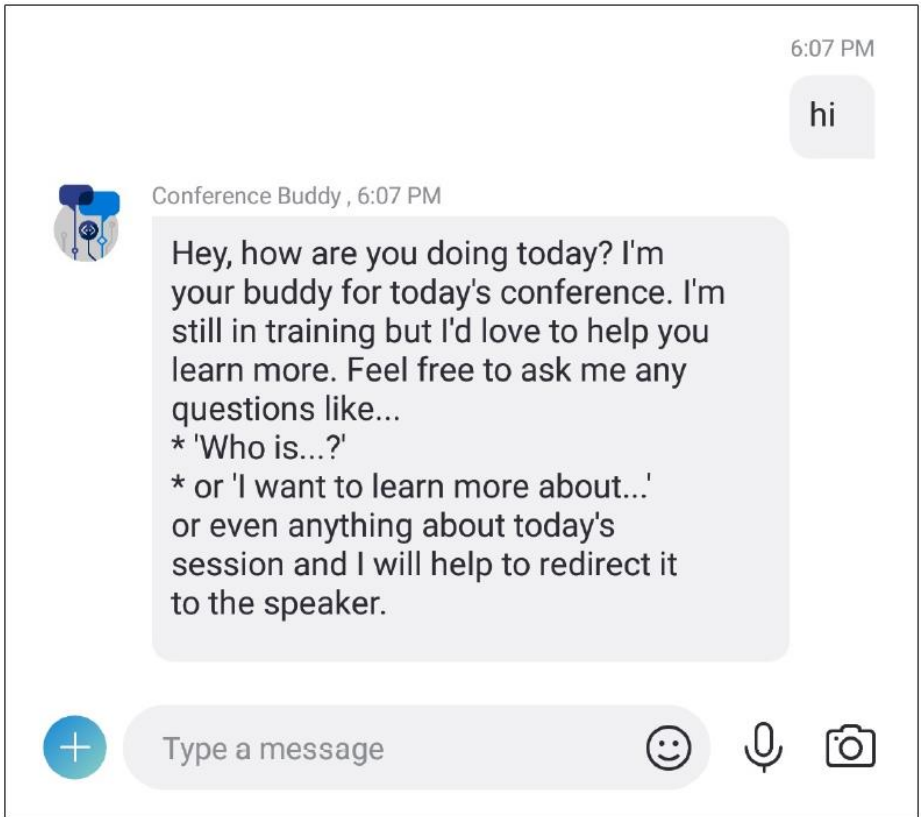


Imagen 1-6. Saludo del bot Conference Buddy

Conference Buddy realiza muchas acciones que son indicativas de los buenos principios de diseño para iniciar el diálogo:

- Primero el bot te saluda: “Hola, ¿qué tal estás?”
- Y después se presenta: “Hoy seré tu compañero en la conferencia. Todavía estoy aprendiendo pero creo que podré ayudarte”.
- Proporciona detalles de todo lo que sabe hacer: “No dudes en preguntarme cualquier duda...”
- Por último, se ofrecen sugerencias acerca del formato que puede usarse para expresar las preguntas: “Quién es...” “Quiero obtener más información sobre...” para facilitar el procesamiento.

Conference Buddy envía el mensaje que recibe del usuario a LUIS para determinar la intención del mensaje. A continuación, selecciona la tarea adecuada del bot en el Cerebro del bot para lanzarla a través de una publicación HTTP y procesar el mensaje. Analizaremos más detalles cuando tratemos el flujo de conversación a modo de ejemplo en la siguiente subsección.

## Cerebro del bot

El cerebro del bot es la lógica o la inteligencia comercial que alimenta el bot Conference Buddy. En [Imagen 1-5](#) hemos resumido las tareas actuales del bot dentro del cerebro del bot:

- Tarea de preguntar quién
- Tarea de obtener más información
- Tarea de responder preguntas

El cerebro del bot es un conjunto de tareas inteligentes del bot donde cada tarea del bot es una función para completar una única tarea y es independiente de otras tareas del bot. Se trata de una de las formas en que hemos implementado los microservicios en nuestra arquitectura. Si una tarea del bot devuelve un mensaje de error, será más sencillo depurarlo y reparar el origen del problema sin afectar al resto de tareas o componentes del bot en la arquitectura.

Otro ejemplo de implementación de microservicios es cómo conseguimos que el bot de conferencia que gestiona los eventos de mensajes estuviera separado de la lógica. De esta forma, cada componente solo se preocupa de llevar a cabo sus propias funciones. Esto facilita la optimización del rendimiento y el escalado. Una analogía del funcionamiento de los microservicios sería imaginar una planta de fabricación de coches donde las distintas piezas provienen de China, Japón, Estados Unidos, etc. Desde el punto de vista de los empleados de la cadena de montaje, no deben preocuparse de los trabajos internos específicos de cada pieza, sino que solo deben preocuparse por montar cada pieza en el lugar adecuado, conectarla y asegurarse de que el vehículo funcione en su totalidad. Si una pieza no funciona, suele quedar aislada y será necesario reemplazarla o repararla sin tener que desmontar todo el vehículo.

Lo que permite al bot de conferencia hablar con el cerebro del bot es el contrato de datos. El contrato de datos se compone de un objeto de solicitud y un objeto de respuesta, y especifica el formato y las propiedades de lo que debe incluir la pregunta y la respuesta. El contrato de datos es lo que permite realizar la abstracción dentro del bot Conference Buddy. Sin el contrato de datos, el bot de conferencia no podría invocar el cerebro del bot ni acceder a las tareas del bot para procesar la respuesta.

Además, el cerebro del bot puede reutilizarse y otros bots y otras aplicaciones pueden usarlo. El cerebro del bot también puede desarrollarse y aumentar en inteligencia añadiendo nuevas tareas para el bot.

En la siguiente sección explicaremos cómo enseñar nuevas habilidades al cerebro del bot.

## Tarea del bot

La tarea del bot es una función de la lógica comercial que puede beneficiarse de varias API de servicios cognitivos para procesar los mensajes de los usuarios. La tarea del bot es una función dentro de la recopilación del cerebro del bot. Cada tarea del bot puede desplegarse en el mismo servicio web, o en servicios web separados, y pueden escalarse independiente entre sí.

Por ejemplo, la tarea “Preguntar quién” utiliza dos API de servicios cognitivos: búsqueda en Internet de Bing y Búsqueda de imagen de Bing. Después, combina los resultados en un objeto de respuesta y lo envía de vuelta al bot de conferencia. A continuación, el bot de conferencia crea una tarjeta gráfica rica que será devuelta al usuario. Dado que todas las tareas del bot en Conference Buddy invocan la API de búsqueda en Internet de Bing, nuestro chatbot tendrá acceso inmediato a todos los conocimientos globales online a través de una búsqueda inteligente que ofrecerá resultados relevantes. Además, la tarea del bot "Más información" activa la búsqueda personalizada en Bing, que le permite usar las mismas funcionalidades de la API que limita la búsqueda de Bing a un número restringido de sitios web. En nuestro ejemplo del bot Conference Buddy, hemos limitado los sitios web a temas relacionados con la conferencia.

Las tareas del bot se encuentran en una aplicación única de Azure Functions pero existen como funciones separadas. Azure Functions es una solución para ejecutar fácilmente pequeños segmentos de código o funciones en el *cloud*. Puede escribir solo el código que necesita para el problema actual, sin preocuparse por toda la aplicación o la infraestructura necesarias para ejecutarlo. Cada tarea del bot también puede enviar el mensaje a Azure Search Questions Store para ser utilizado por el panel.

### NOTA

Nosotros le proporcionamos un código de muestra para ayudarle a empezar a usar cada uno de las pautas comunes y otros escenarios incluidos en [Bot Builder Samples Repository en Github](#).

## Flujo de conversación: Un ejemplo del bot Conference Buddy en acción

Para hacerse una idea de cómo funciona el bot Conference Buddy vamos a examinar un flujo de conversación típico:

1. El usuario invoca el bot de conferencia enviando el primer mensaje.
2. El bot de conferencia responde con un saludo y un resumen introductorio de lo que puede hacer.
3. El usuario hace una pregunta, por ejemplo, “¿Quién es Lili Cheng?”
4. El bot de conferencia envía el mensaje a LUIS para determinar la intención del mensaje: LUIS analiza el mensaje y, en nuestro ejemplo, devuelve “Esta es una tarea de preguntar quién”.
5. A continuación, el bot de conferencia selecciona la tarea adecuada del bot dentro del cerebro del bot para que lanzará a través de una publicación HTTP. En nuestro ejemplo, la tarea “Preguntar quién” hará lo siguiente:
  - a. Enviar la cadena a Bing Web Search y obtener resultados.
  - b. Enviar la cadena a Bing Image Search en paralelo.
  - c. Combinar la imagen y el texto en un contrato de datos/objeto de respuesta que comprenda el bot de conferencia.
6. El bot de conferencia envía una tarjeta gráfica a modo de resultados al usuario.
7. El bot de conferencia envía los resultados a Azure Search para archivarlos y que puedan usarse en el panel.
8. El usuario puede hacer clic en la tarjeta para obtener más información del origen del artículo.

**Imagen 1-7** ilustra la tarjeta de respuesta “¿Quién es?” para “Lili Cheng”.

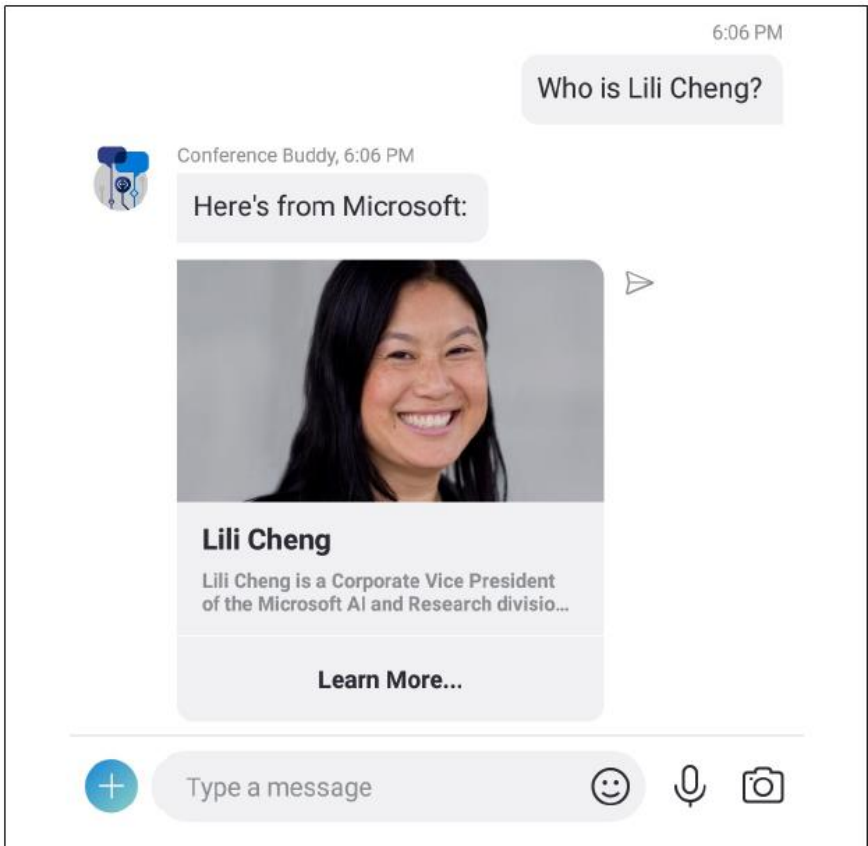


Imagen 1-7. Tarjeta “¿Quién es?”

Ahora vamos a hacer una demostración de la tarea “Más información” para ilustrar todo el proceso:

1. Supongamos que el usuario pregunta “Quiero aprender más sobre Azure Machine Learning”.
2. El bot de conferencia envía el mensaje a LUIS para determinar la intención del mensaje: LUIS analiza el mensaje y, en nuestro ejemplo, devuelve “Esta es una tarea de obtener más información”.
3. A continuación, el bot de conferencia selecciona la tarea apropiada del bot para lanzarla a través de una publicación HTTP y procesar el mensaje: En nuestro ejemplo, la “Tarea de obtener más información” lanzará Text Analytics para extraer frases clave y enviar solicitudes paralelas a:

- a. **Video Indexer:** Video Indexer es un servicio cognitivo que emplea la transcripción del vídeo, la desglosa en palabras clave, anota el vídeo, analiza el sentimiento y modera el contenido. Puede subir vídeos específicos relacionados con la sesión y reproducirá el vídeo en el momento oportuno cuando el orador trate la palabra clave introducida.
- b. **Bing Custom Search:** Habilita el complemento Bing Search en un número limitado de páginas web para aumentar la relevancia y la velocidad de los resultados. En el caso del bot Conference Buddy, hemos incluido páginas web que solo tratan los temas de conferencia.
- c. **Bing Web Search:** Bing Web Search se activa si el indexador de vídeo anterior y la búsqueda personalizada con Bing no han generado resultados.

Ahora pensemos en algunas consideraciones de diseño y analicemos la arquitectura del bot.

## Consideraciones de diseño del bot Conference Buddy

A un nivel alto, Un bot es como cualquier otra aplicación o página web, por lo que se aplican las mismas consideraciones de diseño que para crear una buena interfaz de usuario (IU) o una buena experiencia de usuario (EU).

A la hora de crear nuestro bot Conference Buddy, hemos tenido en cuenta las siguientes cuestiones:

### *Identificar el propósito*

¿Cuál es el propósito de crear el bot? ¿Cuál es el objetivo de los usuarios a la hora de interactuar con el bot? En este caso, queremos dar a los asistentes de la conferencia una plataforma para hacer preguntas y hacer colaborar al orador.

### *Pautas generales para diseñar el bot*

¿Qué pautas debemos seguir a la hora de diseñar el bot? El bot Conference Buddy es un ejemplo de pauta de bot de Base de conocimientos: un bot que encuentra y devuelve la información que ha solicitado un usuario, con una opción de escalar la pregunta a un agente humano. Existen otros ejemplos y otras pautas comunes para los bots:

- **Finalización y automatización de tareas:** Esta pauta permite que el usuario complete una tarea o una serie de tareas específicas sin la ayuda de una persona humana. Un ejemplo podría ser un bot de Restablecimiento de contraseña que puede guiar al usuario a través del proceso de cambio de la contraseña y liberar al personal de atención al cliente para que pueda centrarse en tareas más complejas. Otro ejemplo podría ser un bot asistente de RR.HH. que pueda rellenar una solicitud de empleado para cambiar sus apellidos y actualizar los registros y el perfil de empleado.



- **Bot para web:** Este bot envía al usuario a un explorador para completar una tarea y después reanuda la conversación con el usuario después de completar la tarea. Algunos ejemplos de uso serían las gestiones de seguridad, autenticación y autorización. Después de verificar al usuario, el bot accederá a los datos personales (con la aprobación del usuario) y proseguirá con la tarea que está realizando. Un ejemplo sería un chatbot de agente de seguros virtual que verifica un cliente existente y después ayuda al cliente a actualizar su plan o hacer cambios en su póliza.
- **Derivación a humano:** Esta pauta realiza una transición fluida a una persona humana cuando identifica un escenario que requiere la intervención humana.
- **Bots en aplicaciones:** Esta opción ayuda al usuario a navegar por aplicaciones más complejas y deriva el caso a una persona humana cuando es necesario. Un ejemplo sería una aplicación de usuario de atención al cliente que gestiona una primera respuesta cuando el usuario interactúa con el bot.
- **Bots en sitios web:** Esta pauta ayuda a los usuarios a navegar por páginas web complejas y a buscar información más rápidamente.

## Plataforma de mensajería

¿En qué plataforma de mensajería estará ubicado el bot? ¿De qué plataformas y dispositivos se ocupan nuestros usuarios? Existen canales de mensajería muy populares como Skype, Facebook Messenger, Slack, Kik y otros, o puede crear una capa de mensajería personalizada en una página web o una aplicación móvil existente.

La clave es averiguar dónde pasa el mayor tiempo su audiencia objetivo. ¿Tiene una plataforma de juegos muy popular y desea introducir un bot de recompensa de juego? ¿Un pequeño negocio con seguimiento en los medios sociales? ¿Un gran banco con una aplicación móvil muy popular? La localización de su bot también deberá estar vinculada al motivo específico por el que lo está creando.

Para llegar al máximo número posible de miembros de la audiencia, hemos decidido hacer que nuestro bot Conference Buddy tenga varios canales. Para ello, tendría que desarrollar un complemento especial para cada fuente que se ocupe del protocolo específico entre la fuente de información y su sistema. Microsoft Azure Bot Service Framework le permite conectarse con más de un canal de mensajería y recibir la información en formato estándar, independientemente de su fuente.

**Imagen 1-8** Muestra la pantalla de Microsoft Azure Bot Service donde podrá añadir nuevos canales con tan solo unos clics del ratón.<sup>10</sup>

---

<sup>10</sup> Cómo conectar un bot a canales

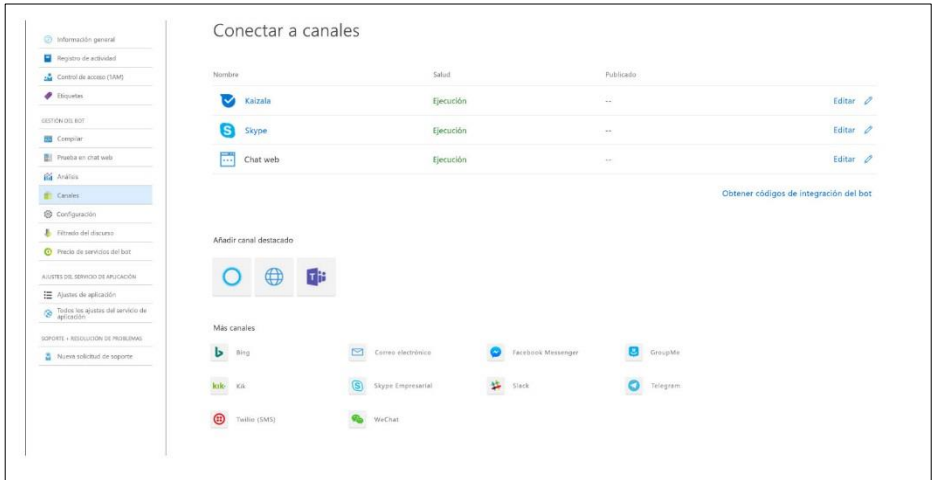


Imagen 1-8. Varias conexiones con canales

## Arquitectura general del bot

¿Cómo transformará su bot la información a un formato estándar que pueda utilizarse para su procesamiento? ¿Y cómo devolverá la información de forma que pueda utilizarse en el canal? Para nuestro bot Conference Buddy, implemente un contrato de datos para describir de forma abstracta los datos que deben intercambiarse. Un contrato de datos define qué datos se han serializado (transformado a XML) para intercambiarlos de forma precisa y para cada parámetro o tipo de respuesta.

### NOTA

Para los principios generales de diseño del bot y recomendaciones específicas, consulte la [sección Principios de diseño del bot](#) en la documentación de Microsoft Azure Bot Service.

## Detalles de arquitectura del bot Conference Buddy

Ahora vamos a ver en detalle la arquitectura del bot Conference Buddy y descubriremos el poder del chatbot.

### Diálogo principal

Mientras que una aplicación tradicional empieza con una pantalla principal y los usuarios pueden navegar y volver al inicio, los bots tienen el *Diálogo principal*. El Diálogo principal guía todo el flujo de conversación. Desde una perspectiva de la IU, cada diálogo actúa como una nueva pantalla. De esta

forma, los diálogos ayudan al desarrollador a separar de forma lógica las distintas áreas de la funcionalidad completa del bot.

Para el bot Conference Buddy, cada diálogo invoca al siguiente dependiendo de lo que el usuario escriba y cuál sea su intención. Este fenómeno recibe el nombre de *diálogo de cascada*. Un diálogo de cascada es un tipo de diálogo que permite al bot guiar a los usuarios fácilmente por una serie de tareas o bien recopilar información. Las tareas se implementan a modo de conjunto de funciones, donde los resultados de la primera función se envían como información a la siguiente función, y así sucesivamente. Cada función suele representar un paso del proceso global. A cada paso, el bot pide al usuario que introduzca datos, espere una respuesta y después pasa el resultado al siguiente paso.

En el caso de nuestro bot Conference Buddy, si el usuario escribe:

```
"¡Hola, colega!"
```

El diálogo principal enviará la cadena a LUIS y esperará una respuesta. LUIS evaluará la cadena y la mandará de vuelta un objeto JSON con los resultados. Para cada intención, LUIS facilita una puntuación de certeza, destaca la intención de puntuación superior e identifica a las entidades de la consulta. El código siguiente muestra un ejemplo de respuesta:

```
{
  "query": "Hello there, buddy",
  "topScoringIntent": {
    "intent": "Greeting",
    "score": 0.9887482
  },
  "intents": [
    {
      "intent": "Greeting",
      "score": 0.9887482
    },
    {
      "intent": "who",
      "score": 0.04272597
    },
    {
      "intent": "learnmore",
      "score": 0.0125702191
    }
  ],
  "entities": [
    {
      "entity": "buddy",
      "type": "Person",
      "startIndex": 20,
      "endIndex": 24,
      "score": 0.95678144
    }
  ]
}
```

Cuando LUIS devuelve la intención como un “saludo”, el diálogo principal procesa la función "Intención de saludo". Esta función muestra el diálogo de saludo que, en nuestro ejemplo, no tiene que invocar una tarea del bot. El control permanecerá en el diálogo de saludo hasta que el usuario introduzca otro texto distinto. Cuando el usuario responde, el diálogo de saludo se cierra y el diálogo principal retoma el control.

Ahora vamos a analizar el siguiente código de muestra del diálogo principal para ver cómo se procesan las otras intenciones:

```
public Task StartAsync(IDialogContext context)
{
    context.Wait(MessageReceivedAsync);
    return Task.CompletedTask;
}
private async Task MessageReceivedAsync(IDialogContext context,
    IAwaitable<object> result)
{
    try
    {
        var activity = await result as Activity;
        string message = WebUtility.HtmlDecode(activity.Text);
        if (string.IsNullOrEmpty(message) == true)
        {
            return;
        }
        // Handle the explicit invocation case in Skype
        string channelId = GetChannelId(activity);
        if (channelId == "skype" &&
            message.StartsWith(activity.Recipient.Name) == true)
        {
            message =
                message.Substring(activity.Recipient.Name.Length).Trim( );
        }
        else if (channelId == "skype" &&
            message.StartsWith
                ("@" + activity.Recipient.Name) == true)
        {
            message =
                message.Substring
                    (activity.Recipient.Name.Length + 1).Trim( );
        }
        // Handle intents
        LUISResult luisResult = await GetEntityFromLUIS(message);
        string intent =
            luisResult.intents?.FirstOrDefault()?.intent ?? string.Empty;
        string[ ] entities =
            luisResult.entities?.Select
            (e => e.entity)?.ToArray( ) ?? new string[0];
        if (intent == "greeting")
        {
            await ProcessGreetingIntent(context, message);
        }
        else if (intent == "who")
        {
            await ProcessQueryIntent
                (context, activity, BotTask.AskWho, message, entities);
        }
    }
}
```

```

else if (intent == "learnmore")
{
    await ProcessQueryIntent
(context, activity, BotTask.AskLearnMore, message, entities);
}
else
{
    await ProcessQueryIntent(
        context, activity,
        BotTask.AskQuestion, message, entities);
}

```

El diálogo principal no se invoca a no ser que el usuario introduzca un mensaje. Cuando el bot Conference Buddy recibe el primer mensaje, realizamos una manipulación especial en el código para los mensajes que provienen del canal Skype.

Ya hemos explicado qué ocurre cuando LUIS devuelve una intención de salutación. En nuestro ejemplo de chatbot, anticipamos tres otras posibles intenciones de LUIS:

- Si la intención es “Quién”, el diálogo principal publica la pregunta en la tarea del bot "Preguntar quién".
- Si la intención es “Más información”, el diálogo principal publica la pregunta en la tarea del bot “Más información”.
- Para el resto de intenciones, el diálogo principal envía el texto al mensaje “Hacer pregunta”.

En este punto, el diálogo principal asume el control para la tarea del bot apropiada.

## Capa de abstracción del cerebro del bot

La capa de abstracción gestiona la llamada de publicación a una tarea del bot dentro del cerebro del bot. Aquí es donde queda claro el beneficio de la implementación de los microservicios. El diálogo principal ha gestionado el mensaje y LUIS ha procesado la intención. A este nivel, el bot ejecuta la tarea del bot pertinente.

Ahora vamos a explorar el código:

```

private static async
Task<string>ProcessQueryIntent(IDialogContext context,
    Activity activity, BotTask task, string query, string [ ] topics)
{
    // Prepare the request to invoke a bot task within the bot brain
    AskQuestionRequest request = new AskQuestionRequest( )
{
    ConversationId = activity.Conversation.Id,

```

```

        Question = query,
        SessionId = SessionId,
        Topics = topics != null ? topics.ToArray() : new string[0],
        UserId = string.IsNullOrEmpty(activity.From.Name)
            == false ? activity.From.Name : activity.From.Id
    };
    // Invoke the bot task to process the request
    AskQuestionResponse askQuestionResponse =
        await
            HttpClientUtility.PostAsJsonAsync
            <AskQuestionResponse>
            (new Uri(BotBrainUrl + task.ToString()), RequestHeaders, request);
    // Handle the response returned from the
    // bot task to be shown as cards depending on channel
    if (askQuestionResponse.Results?.Count() > 0 == true)
    {
        IMessageActivity foundMsg = context.MakeMessage();
        AskQuestionResult result = askQuestionResponse.Results[0];
        if (string.IsNullOrEmpty(result.Source) == false)
        {
            foundMsg.Text = string.Format
            ("Got it. Meanwhile, from {0}:", result.Source);
        }
        else
        {
            foundMsg.Text = "Got it. Meanwhile, here's what I found:";
        }
        await context.PostAsync(foundMsg);
        IMessageActivity cardMessage;
        string channelId = GetChannelId(activity);
        if (channelId == "kaizala")
        {
            cardMessage = await
                GetKaizalaCardMessage(context, request, result);
        }
        else if (channelId == "directline" || channelId == "emulator")
        {
            cardMessage =
                GetAdaptiveCardMessage(context, request, result);
        }
        else
        {
            cardMessage = GetHeroCardMessage(context, request, result);
        }
        await context.PostAsync(cardMessage);
    }
    else if (task != BotTask.AskQuestion)
    {
        IMessageActivity notFoundMsg = context.MakeMessage();
        notFoundMsg.Text =
            "I can't seem to find it.
            Can you rephrase the question and try again?";
        await context.PostAsync(notFoundMsg);
    }

    return "success";
}

```

Lo que resulta importante en esta capa, independientemente de la tarea del bot solicitada, es que la solicitud, la invocación y la repuesta se gestionan de la misma forma.

Contrato de datos invocado

AskQuestionRequest combina el ConversationID, Query, SessionID y UserID para mandarlos a la tarea del bot a través de una publicación HTTP.

La publicación HTTP es la invocación de la tarea del bot dentro del cerebro del bot. Cuando la tarea del bot correspondiente ejecuta la consulta, prepara la respuesta en AskQuestion Response donde la respuesta se gestiona de forma genérica con independencia de la tarea del bot.

Dado que el bot Conference Buddy no es de varios canales, la tarjeta de respuesta se muestra de forma distinta dependiendo del canal; la última parte del código muestra cómo el bot implementa las tarjetas adaptativas.

## El contrato de datos

Sin el contrato de datos la capa de abstracción no existiría. El código del contrato de datos que sigue actúa como el acuerdo formal entre el bot y el cerebro del bot y describe de forma abstracta los datos que van a intercambiarse.

Pasemos a explorar el código y ver los detalles de AskQuestionRequest, que especifica los detalles que se enviarán con cada consulta, y de AskQuestionResponse, que especifica los detalles para cada respuesta con independencia de la tarea que ejecute el bot:

```
namespace ConferenceBuddy.Common.Models
{
    [DataContract]
    public class AskQuestionRequest
    {
        /// <summary>
        /// The session identifier
        /// </summary>
        [DataMember(Name = "sessionId")]
        public string SessionId { get; set; }
        /// <summary>
        /// The conversation identifier
        /// </summary>
        [DataMember(Name = "conversationId")]
        public string ConversationId { get; set; }
        /// <summary>
        /// The user identifier
        /// </summary>
        [DataMember(Name = "userId")]
        public string UserId { get; set; }
        /// <summary>
        /// The text of the question
        /// </summary>
        [DataMember(Name = "question")]
        public string Question { get; set; }
        /// <summary>
        /// The topics of the question
        /// </summary>
        [DataMember(Name = "topics")]
        public string [] Topics { get; set; }
    }
}
```

```

[DataContract]
public class AskQuestionResponse
{
    /// <summary>    /// The unique id of the response
    /// </summary>
    [DataMember(Name = "id")]

    public string Id { get; set; }

    /// <summary>
    /// The results of the response
    /// </summary>
    [DataMember(Name = "results")]
    public AskQuestionResult [] Results { get; set; }
}
[DataContract]
public class AskQuestionResult
{
    /// <summary>
    /// The title of the result
    /// </summary>
    [DataMember(Name = "title")]

    public string Title { get; set; }

    /// <summary>
    /// The answer of the result
    /// </summary>
    [DataMember(Name = "answer")]
    public string Answer { get; set; }

    /// <summary>
    /// The image url of the result
    /// </summary>
    [DataMember(Name = "imageUrl")]
    public string ImageUrl { get; set; }

    /// <summary>
    /// The source of the result
    /// </summary>
    [DataMember(Name = "source")]
    public string Source { get; set; }

    /// <summary>
    /// The url of the result
    /// </summary>
    [DataMember(Name = "url")]
    public string Url { get; set; }

    /// <summary>
    /// The url display name of the result
    /// </summary>
    [DataMember(Name = "urlDisplayName")]
    public string UrlDisplayName { get; set; }
}
}

```

El contrato de datos permite separar las funciones entre cómo se procesa una consulta y cómo se genera la respuesta. Piense en el contrato de datos como si fuese el transportista. Desde la perspectiva del mensajero, los detalles específicos



del contenido en la carta/paquete es irrelevante. Lo que importa es el formato de las direcciones “Para” y “De” para que la entrega se realice en la ubicación correcta.

Si tuviéramos que hacer distintas llamadas HTTP para cada tarea del bot, el bot Conference Buddy sería complejo y difícil de construir, probar, implementar y escalar. En la siguiente sección, veremos cómo la implementación de los microservicios facilita el desarrollo de la inteligencia del cerebro del bot y cómo enseña nuevas habilidades a nuestro Conference Buddy.

## Cómo añadir la inteligencia *plug and play* a su bot

Podemos enseñar a nuestro Conference Buddy nuevas habilidades desarrollando la inteligencia del cerebro del bot. Hasta ahora, hemos creado un bot Conference Buddy que tiene tres tareas principales como bot:

- Tarea de preguntar quién
- Tarea de obtener más información
- Tarea de responder preguntas

Hemos creado la arquitectura de Conference Buddy de forma flexible para que un desarrollador pueda añadir fácilmente más tareas del bot. Ahora vamos a ampliar el escenario de nuestro bot Conference Buddy. Imaginemos que la conferencia va a retransmitirse en todo el mundo y los miembros de la audiencia proceden de distintos países y hablan idiomas diferentes, mientras que el orador solo habla inglés. Es posible que quiera añadir una nueva tarea para que el bot pueda gestionar las preguntas en distintos idiomas y traducir la respuesta al inglés para que el orador la entienda.

Para nuestro bot, haremos una llamada adicional de servicios cognitivos: Microsoft Translator. Se trata de un servicio de traducción automática que admite más de 60 idiomas. El desarrollador envía el texto de origen al servicio con un parámetro que indica el idioma de destino, y el servicio envía el texto traducido para el cliente o la aplicación web puedan usarlo.

El texto traducido ahora puede usarse con los otros servicios cognitivos que hemos usado hasta ahora, como el análisis de texto o la búsqueda en web de Bing.

Para hacer una llamada a un nuevo servicio cognitivo, tiene que iniciar sesión en el portal de Azure. [Esta Guía de inicio rápido](#) le explica cómo editar el código del bot

y cómo usar las funciones de Azure para invocar varias API. En el código de muestra que sigue, mostramos cómo añadir la nueva tarea del bot de traducción. Ahora vamos a explorar el código:

```
[FunctionName("AskQuestion")]
public static async Task<HttpResponseMessage>
Run(
    [HttpTrigger(AuthorizationLevel.Function, "post", Route =
        "AskQuestion")]HttpRequestMessage request,
    [Table("Session", Connection =
        "AzureWebJobsStorage")]ICollector<SessionTableEntity> sessionTable,
    TraceWriter log)
{
    MediaTypeHeaderValue contentType = request.Content.Headers.ContentType;

    // Check if content type is empty
    if (contentType == null)
    {
        return request.CreateResponse
            (HttpStatusCode.BadRequest, "Missing content-type from header.");
    }
    else if (contentType.MediaType.Contains("application/json") == false)
    {
        return request.CreateErrorResponse
            (HttpStatusCode.UnsupportedMediaType,
            string.Format("Request's content type ({0}) is not supported.",
            string.Join(", ", contentType.MediaType)));
    }

    // Read content from request
    AskQuestionRequest requestBody = await
        request.Content.ReadAsAsync<AskQuestionRequest>();

    // Verify content contains a valid image uri
    if (string.IsNullOrEmpty(requestBody.Question) == true)
    {
        return request.CreateResponse(HttpStatusCode.BadRequest,
            "Question is missing from the request content.");
    }
    else if (string.IsNullOrEmpty(requestBody.SessionId) == true)
    {
        return request.CreateResponse(HttpStatusCode.BadRequest,
            "Session id is missing from the request content.");
    }
    // Translate question
    requestBody.Question = await
        ServicesUtility.Translator.TranslateTextAsync(requestBody.Question);

    // Answer the question
    AskQuestionResponse response =
        await AnswerQuestion(requestBody, sessionTable);

    // Return request response with result and 200 OK
    return request.CreateResponse(HttpStatusCode.OK, response); }

public static async
Task<AskQuestionResponse> AnswerQuestion(AskQuestionRequest request,
    ICollector<SessionTableEntity> sessionTable)
```

```

{
    // Get unique identifier
    string id = Guid.NewGuid().ToString();
    DateTime timestampUtc = DateTime.UtcNow;

    // Run keyphrases extraction
    request.Topics = await ServicesUtility.GetTopics
        (request.Question, request.Topics);

    // Run search services
    string queryWithTopics = request.Topics?.Count() > 0 ?
        string.Join(" ", request.Topics).Trim() : request.Question;

    Task<BingWebSearchResult> bingWebSearchTask =
        ServicesUtility.BingSearch.SearchWebAsync
        (query: request.Question, count: SettingsUtility.MaxResultsCount);
    Task<BingWebImagesResult> bingWebImagesTask =
        ServicesUtility.BingSearch.SearchImagesAsync
        (query: request.Question, count: SettingsUtility.MaxResultsCount);
    await Task.WhenAll(bingWebSearchTask, bingWebImagesTask);

    BingWebSearchResult bingWebSearchResult = bingWebSearchTask.Result;
    BingWebImagesResult bingWebImagesResult = bingWebImagesTask.Result;

    // Process results
    AskQuestionResponse response = new AskQuestionResponse()
    {
        Id = id,
        Results = new AskQuestionResult[0]
    };

    if (bingWebSearchResult.WebPagesResult?.Values?.Count() > 0)
    {
        response.Results =
            ServicesUtility.GetAskQuestionResults(bingWebSearchResult);
    }

    if (response.Results.Any(r => string.IsNullOrEmpty(r.ImageUrl))
        == true) == true && bingWebImagesResult?.Values?.Count()
        > 0 == true)
    {
        response.Results =
            ServicesUtility.AddImageToResults(response.Results,
                bingWebImagesResult);
    }

    // Upload search document
    await
        ServicesUtility.UploadDocumentToSearchService
        (SettingsUtility.AzureSearchIndexName,
            new SessionSearchDocument
            (id, timestampUtc, "AskQuestion", request, response));

    // Write to session table
    sessionTable.Add(new SessionTableEntity
        (id, timestampUtc, "Question", request, response));

    return response;
}

```

En la primera parte del código, la función AskQuestion lee el contenido de la solicitud y traduce la pregunta a inglés usando el traductor. A continuación, extrae las frases clave usando Text Analytics y envía la consulta a Bing Web Search y Bing Image Search para crear una tarjeta para la respuesta. Las frases clave se envían a Azure Search para alimentar el análisis del bot y el panel. En este ejemplo, no traducimos la respuesta de vuelta al idioma original, pero esa podría ser una opción para otras implementaciones.

Ahora que ya hemos añadido correctamente una nueva tarea del bot, seguimos desarrollando la inteligencia del cerebro del bot para añadir más habilidades como incorporar otras API de visión, discurso y otros elementos a través de nuestros servicios cognitivos. Ahora exploraremos el panel de Conference Buddy.

## **Cómo crear una aplicación comercial para obtener conocimientos del bot: el panel de Conference Buddy**

El panel de Conference Buddy forma parte del bot Conference Buddy. El panel actúa como el repositorio de preguntas y respuestas para que puedan explorarlas los asistentes de la conferencia y el orador. El panel de Conference Buddy (consulte [Imagen 1-9](#)) hace lo siguiente:

- Muestra las preguntas de todos los miembros de la audiencia en tiempo real
- Permite al orador buscar, ordenar o filtrar rápidamente los resultados por Sesión, Habilidades del bot o Tema para ver las preguntas enviadas relevantes

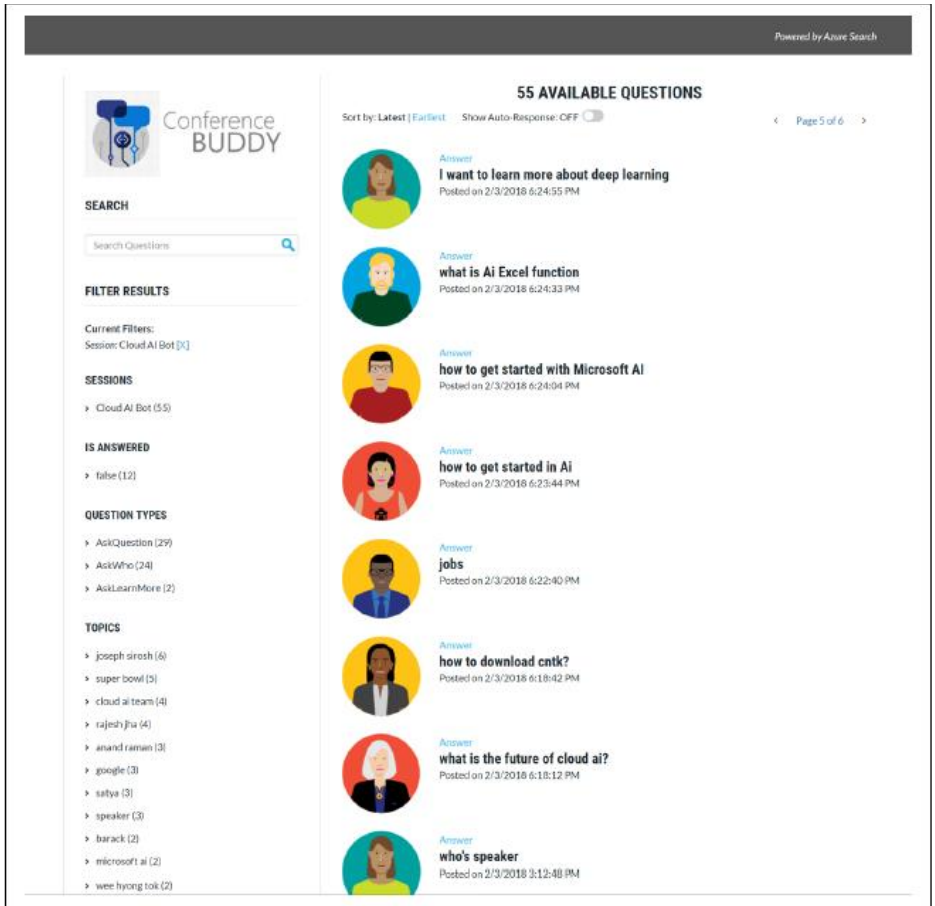


Imagen 1-9. La aplicación del panel de Conference Buddy

Hemos creado el panel usando ASP.NET Core MVC, que es un entorno rico para crear aplicaciones web y API usando el patrón de diseño Model-View-Controller. **Puede encontrar más información aquí** para conocer los pasos de creación de una aplicación web similar a nuestro panel de Conference Buddy.

Muchas aplicaciones web necesitarán una función de búsqueda para el contenido de la aplicación. Tener una API de búsqueda fácil de usar puede ser una gran ventaja para los desarrolladores. Nosotros hemos integrado Azure Search en nuestro panel de Conference Buddy para buscar las preguntas que se están realizando. Azure Search es una API Search-as-a-Service muy sencilla que permite a los desarrolladores integrar una experiencia de búsqueda sofisticada en las páginas web y x`las aplicaciones móviles sin tener que preocuparse por las complejidades de la búsqueda de texto completo y sin tener que implementar, mantener o gestionar infraestructuras.

Para el panel de Conference Buddy, Azure Search hace lo siguiente:

- Alimenta la funcionalidad de búsqueda
- Indexa las frases clave extraídas para aparecer como temas
- Permite filtrar los resultados y ordenarlos para crear referencias simples.

El código de muestra siguiente muestra cómo lanzar Azure Search:

```
public class SessionSearchService
{
    private ISearchIndexClient IndexClient;

    public SessionSearchService()
    {
        string searchServiceName =
            ConfigurationManager.AppSettings["SearchServiceName"];
        string index =
            ConfigurationManager.AppSettings["SearchServiceIndexName"];
        string apiKey =
            ConfigurationManager.AppSettings["SearchServiceApiKey"];

        SearchServiceClient searchServiceClient = new
            SearchServiceClient
            (searchServiceName, new SearchCredentials(apiKey));
        this.IndexClient =
            searchServiceClient.Indexes.GetClient(index);
    }

    public SessionSearchService()
    {
        string searchServiceName =
            ConfigurationManager.AppSettings["SearchServiceName"];
        string index =
            ConfigurationManager.AppSettings["SearchServiceIndexName"];
        string apiKey
            = ConfigurationManager.AppSettings["SearchServiceApiKey"];

        SearchServiceClient searchServiceClient = new
            SearchServiceClient
            (searchServiceName, new SearchCredentials(apiKey));
        this.IndexClient = searchServiceClient.Indexes.GetClient(index);
    }

    public async Task<DocumentSearchResult>
        SearchAsync(string searchText, string sessionFacet,
            string topicsFacet, string skillFacet,
            string isAnsweredFacet, int currentPage, int
            numResultsPerPage, bool sortByLatest)
    {
        // Execute search based on query string
        Try
        {
            if (string.IsNullOrEmpty(searchText) == true)
            {
                searchText = "*";
            }

            SearchParameters sp = new SearchParameters( )
```

```

{
    SearchMode = SearchMode.Any,
    Top = numResultsPerPage,
    Skip = currentPage * numResultsPerPage,
    // Limit results
    Select = new List<String>( )
    {
        "id", "userId", "sessionId", "question",
        "skill", "topics", "timestampUtc", "answerTitle",
        "answerContent", "answerImageUrl", "answerSource",
        "answerUrl", "answerUrlDisplayName", "isAnswered"
    },
    // Add count
    IncludeTotalResultCount = true,
    // Add facets
    Facets = new List<String>()
    { "sessionId,count:0", "topics,count:20",
      "skill,count:0", "isAnswered,count:0" },
      MinimumCoverage = 75
    };

    string orderBy = sortByLatest == true ? "desc" : "asc";

    sp.OrderBy = new List<String>( ) { "timestampUtc " + orderBy };

    // Add filtering
    IList<string> filters = new List<string>();

    if (string.IsNullOrEmpty(sessionFacet) == false)
    {
        filters.Add(string.Format
            ("sessionId eq '{0}'", sessionFacet));
    }

    if (string.IsNullOrEmpty(skillFacet) == false)
    {
        filters.Add(string.Format("skill eq '{0}'", skillFacet));
    }

    if (string.IsNullOrEmpty(topicsFacet) == false)
    {
        filters.Add(string.Format
            ("topics/any(kp: kp eq '{0}')), topicsFacet));
    }

    if (string.IsNullOrEmpty(isAnsweredFacet) == false)
    {
        filters.Add(string.Format
            ("isAnswered eq {0}", isAnsweredFacet));
    }

    sp.Filter =
        filters.Count() > 0 ?
        string.Join(" and ", filters) : string.Empty;

    return await
        this.IndexClient.Documents.SearchAsync(searchText, sp)
}
catch (Exception ex)
{

```

```
Console.WriteLine
  ("Error querying index: {0}\r\n", ex.Message.ToString());
}
return null;
}
```

## Conocimientos del bot: cómo instrumentar a su bot

La mayoría de aplicaciones web y bots querrán analizar el uso y otras estadísticas del bot. Estos análisis también pueden ayudar a detectar y diagnosticar excepciones y problemas de rendimiento de la aplicación. Azure Bot Service proporciona análisis del bot que son una extensión de Azure Application Insights. Application Insights proporciona *datos de nivel de servicio y de instrumentación*, tales como tráfico, latencia e integraciones. Bot Analytics genera informes *de nivel de conversación* sobre el usuario, los mensajes y los datos del canal. Bot Analytics le ofrece todas las ventajas de los conocimientos sin tener que escribir ni una sola línea de código.

Para activar el análisis en el bot (consulte [Imagen 1-10](#)), haga lo siguiente:

1. Instale un paquete de instrumentación pequeño en su aplicación y configure un recurso de Application Insights en el portal de Azure. La instrumentación supervisa tu aplicación y envía datos de telemetría al portal. La aplicación puede ejecutarse en cualquier parte: no tiene que estar alojada en Azure. Siga los pasos que se indican en la guía [“Cómo crear un recurso de conocimiento de aplicación”](#).
2. Abra el bot en el panel. Haga clic en “Configuración” y desplácese hasta la sección “Análisis”.
3. Escriba la información para conectar el bot a Application Insights. Todos los campos son obligatorios.



## Análisis

Habilite el análisis para su bot a través de Azure Application Insights.

Clave de instrumentación de AppInsights ?

Clave de la API de AppInsights ?

ID de la aplicación AppInsights ?

Imagen 1-10. Pantalla de activación del análisis

### Clave de instrumentación de AppInsights

Para encontrar este valor, abra Application Insights, y navegue hasta “Configurar → Propiedades”.

### Clave de la API de AppInsights

Proporcione una clave para la API de Azure App Insights. Aprenda a **generar una nueva clave de API**.

Se necesita un permiso de solo lectura.

### ID de la aplicación AppInsights

Para encontrar este valor, abra Application Insights, y navegue hasta “Configurar → Acceso API”.

### Ver el análisis para el bot

Para ir al análisis, abra el bot en el portal del desarrollador y haga clic en “Analytics”.

Analytics le permite especificar mensajes/datos a través de:

#### Canal

Puede escoger qué canales desea ver en los gráficos. Tenga en cuenta que si un bot no está activado en un canal, no habrá datos de ese canal.

### Periodo de tiempo

El análisis está disponible únicamente para los últimos 90 días. La recopilación de datos se inicia al activar Application Insights.

### Totales globales

Usuarios activos y mensajes enviados.

### Retención

Cuántos usuarios enviaron un mensaje y volvieron, como se indica en **Imagen 1-11**.

RETENCIÓN - % DE USUARIOS QUE HAN ENVIADO MENSAJES DE NUEVO (ÚLTIMOS 10 DÍAS)											
Fecha	Usuarios	Días después									
		1	2	3	4	5	6	7	8	9	10
4/23/2017	0	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
4/24/2017	13	31%	8%	8%	8%	8%	8%	8%	15%	8%	
4/25/2017	43	2%	2%	2%	5%	5%	2%	5%	2%		
4/26/2017	27	4%	4%	4%	4%	4%	4%	4%			
4/27/2017	37	3%	3%	3%	3%	3%	3%				
4/28/2017	22	5%	5%	5%	5%	5%					
4/29/2017	23	9%	9%	9%	9%						
4/30/2017	14	7%	7%	7%							
5/1/2017	17	18%	12%								
5/2/2017	87	9%									

Imagen 1-11. Pantalla de Insights con los usuarios que han enviado mensajes a otros usuarios

El gráfico de usuarios hace un seguimiento de los usuarios que han accedido al bot usando cada canal durante el periodo de tiempo especificado, como se muestra en **Imagen 1-12**.

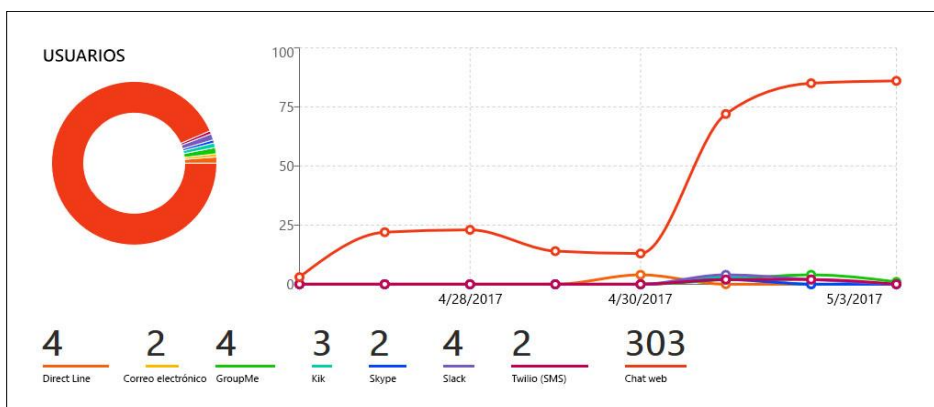


Imagen 1-12. Pantalla de Insights que muestra los usuarios

## Mensajes

El gráfico de mensajes hace un seguimiento de los mensajes que se han enviado usando un canal específico durante el periodo de tiempo especificado (Imagen 1-13).

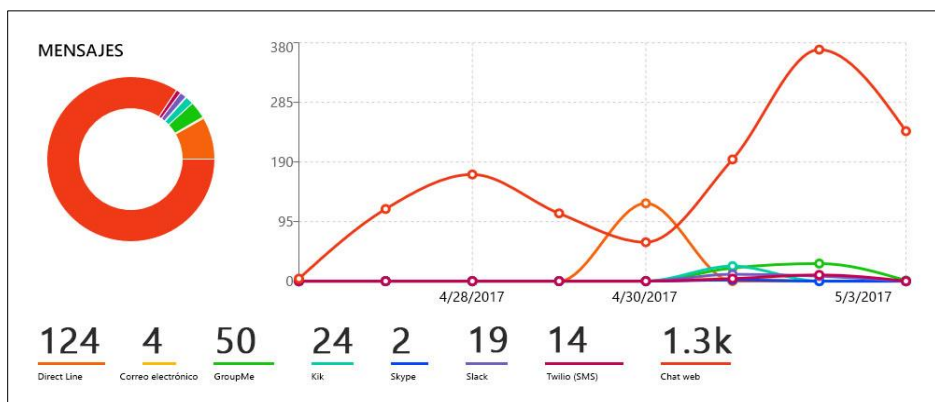


Imagen 1-13. Pantalla de Insights mostrando los mensajes

## Cómo allanar el camino

Nuestro compromiso con la democratización de la IA va más allá de la plataforma de IA en el cloud. En colaboración con Facebook, hemos creado recientemente el formato Open Neural Network Exchange (ONNX, consulte <https://onnx.ai>), un estándar de fuente abierto para representar modelos de deep learning que permite transferir los modelos entre los distintos entornos. Caffe2, PyTorch, Microsoft Cognitive Toolkit (CNTK), Apache MXNet y otras herramientas están desarrollando la compatibilidad con ONNX, y una larga lista de socios como AMD, Nvidia, IBM, Intel y AWS han anunciado la futura compatibilidad. ONNX es el primer paso hacia un ecosistema abierto donde los desarrolladores de IA pueden moverse fácilmente entre herramientas punteras y escoger la combinación que mejor les funcione.

ONNX ofrece una representación común del gráfico de computación y ayuda a los desarrolladores a escoger el entorno de IA más adecuado para su tarea. De esta forma, permite a los autores centrarse en las mejoras innovadoras y a los proveedores de hardware racionalizar las operaciones para sus plataformas.

En la última actualización a Windows 10, los desarrolladores podrán usar la IA para ofrecer experiencias más potentes y atractivas. Los científicos de datos pueden formar y proporcionar modelos ONNX para ser usados por desarrolladores de Windows usando Azure Machine Learning. Esto forma parte de la inversión global de Microsoft en su iniciativa de ofrecer una experiencia de desarrollo extraordinaria a los desarrolladores de IA en Windows.

Microsoft cree en la necesidad de llevar la IA a todos los desarrolladores, en cualquier plataforma, usando cualquier idioma, y con un ecosistema de IA abierto que nos ayude a garantizar que los frutos de la IA serán ampliamente accesibles.

Además de allanar el camino para ofrecer herramientas y recursos para democratizar la IA, también ayudamos a responder las desafiantes preguntas que estas nuevas y potentes tecnologías nos *imponen*. “Acabamos siendo lo que contemplamos”. Damos forma a nuestras herramientas y después nuestras herramientas nos dan forma”.<sup>11</sup>

¿Cómo podemos asegurarnos de que la IA se diseña y usa de forma responsable? ¿Cómo podemos definir principios éticos para proteger a las personas? ¿Cómo debemos regular su uso? ¿Y con qué frecuencia la IA afectará al empleo y los puestos de trabajo? Estas preguntas no pueden ser respondidas solo por los tecnólogos. Se trata de una responsabilidad social que genera discusiones en todos los gobiernos, entornos académicos, negocios, sociedades civiles y otras partes interesadas.

Recientemente hemos publicado un libro, *The Future Computed: Artificial Intelligence and its Role in Society*, que trata los grandes problemas a la hora de regular la IA y el futuro. También destaca que nosotros, en Microsoft, hemos identificado nuestros seis principios éticos para guiar el desarrollo y el uso multidisciplinar de la inteligencia artificial:

### *Equidad*

El primer principio resalta los problemas de la justicia social. ¿Cómo podemos diseñar los sistemas de IA para tratar a todos de forma justa y equilibrada, cuando los datos de formación que usamos pueden estar empañados con suposiciones y prejuicios? Un sistema de IA que ofrece orientación sobre una solicitud de préstamo o empleo, por ejemplo ¿puede diseñarse para hacer las mismas recomendaciones a todas las personas con una situación económica similar o con cualificaciones profesionales parecidas? Como desarrollador, tendrá que conocer cómo pueden introducirse los prejuicios en el sistema, cómo trabajar en la creación de un conjunto de datos representativo y cómo educar a los usuarios finales para que puedan entender que el juicio humano debe complementar las recomendaciones de los sistemas informáticos para contrarrestar sus limitaciones inherentes.

---

<sup>11</sup> Existe un debate permanente sobre quién el autor de esta cita: Marshall McLuhan, Winston Churchill y Robert Flaherty están entre ellos. Visite [este enlace](#) para ver la evolución del debate.

### *Fiabilidad y seguridad*

A medida que somos más dependientes de los sistemas de IA, ¿cómo podemos garantizar que nuestros sistemas operen de forma segura, fiable y uniforme? Consideremos un fallo en un sistema alimentado por IA en un hospital que puede significar literalmente la diferencia entre la vida y la muerte de los pacientes. Debemos realizar diseños y pruebas con la máxima rigurosidad en distintas condiciones e incluir medidas de seguridad para combatir los ciberataques y otras intenciones maliciosas. Algunas veces los sistemas reaccionan de forma inesperada dependiendo de los datos. En Microsoft, tuvimos un ejemplo de conducta inesperada durante la presentación de Tay, un chatbot de conversación en Twitter. Tay era un experimento de IA de conversión que empezó a fallar en sus inicios cuando los usuarios empezaron a introducir comentarios racistas y sexistas que Tay aprendió y reflejó rápidamente. Tuvimos que retirar a Tay en tan solo 24 horas. Los desarrolladores deben enseñar a los usuarios finales cuáles son las conductas previstas en condiciones normales, para que cuando las cosas no salen bien los humanos puedan intervenir rápidamente para reducir al mínimo los daños.

### *Privacidad y seguridad*

Nuestras vidas cada vez están más digitalizadas y, por lo tanto, la privacidad y la seguridad adquieren una mayor relevancia. Este debate abarca más elementos que las propias tecnologías usadas para garantizar la seguridad de los datos: debe incluir regulaciones sobre la utilización de los datos y los motivos para los cuales se usa.

### *Inclusividad*

Queremos asegurarnos de que los sistemas de IA puedan capacitar e implicar a personas de diversas habilidades y niveles de acceso. Actualmente, los servicios alimentados con IA ya ayudan a personas con discapacidades visuales, auditivas y cognitivas. La creación de sistemas que tienen en cuenta el contexto e incorporan inteligencia emocional allanará el camino para conseguir cada vez más capacitación.

### *Transparencia y responsabilidad*

Los diseñadores y desarrolladores de sistemas de IA tienen que crear sistemas con la máxima transparencia. Las personas deben entender cómo se toman las decisiones más importantes para sus vidas. Con responsabilidad, es necesario crear juntas de revisión internas para ofrecer directrices y supervisar los sistemas que están bajo su cargo.

Si “los desarrolladores están escribiendo el script del futuro” como dice Joel Spolsky, Presidente ejecutivo de Stack Overflow, nosotros, como desarrolladores, estamos en el asiento delantero de esta gran conversación. “Cada línea de código es una decisión tomada por el desarrollador. Estamos creando el futuro a partir de esas decisiones”.

En Microsoft, nos hemos comprometido a ofrecer las herramientas, el marco ético y las mejores prácticas para fomentar el desarrollo responsable de los sistemas de IA que enriquecerán las vidas humanas y las innovaciones. Todo ello resolverá nuestros problemas más estresantes de la actualidad y anticipará los que vendrán en el futuro. Por último, es importante recordar que, aunque algunas veces nos atascamos en discusiones sobre las herramientas o los algoritmos más adecuados, el poder real de la IA reside en las ideas y las preguntas que puede predecir. Se trata de la consideración ecologista de cómo crear hábitats sostenibles, la decisión del médico sobre cómo tratar mejor a sus pacientes, la curiosidad como ciudadanos científicos que amplía nuestra concienciación colectiva a los límites externos del universo. La IA tiene el potencial de capacitar las causas humanas más nobles y tan solo es el principio de una transformación tecnológica apasionante.

## Agradecimientos

---

Joseph Sirosh, Wilson Lee y Vinod Anantharaman

## Acerca de los autores

---

**Anand Raman** es jefe de personal de AI Platform y director de AI Ecosystem en Microsoft. Anteriormente trabajó como jefe de personal para Microsoft Azure Data Group en los segmentos de Data Platforms y Machine Learning. En la última década, ha dirigido los equipos de gestión y desarrollo de productos en Azure Data Services, Visual Studio y Windows Server User Experience en Microsoft.

**Wee Hyong Tok** es el director principal de ciencia de datos con el equipo AI Platform en Microsoft. Dirige el equipo de Ingeniería y ciencia de datos para IA en el programa IA para la tierra.

Wee Hyong ha trabajado en muchos puestos diferentes durante su carrera: desarrollador, gestor de programa/productos, científico de datos, investigador y estratega, y su historial de liderazgo para equipos de ingeniería y ciencia de los datos le ha aportado súper poderes únicos que le han permitido ser asesor de IA para muchos clientes.

Wee Hyong ha participado como coautor en muchos libros sobre inteligencia artificial, incluido su primer libro *Predictive Analytics Using Azure Machine Learning* and *Doing Data Science with SQL Server*. Wee Hyong ha cursado un doctorado en ciencias informáticas en la Universidad Nacional de Singapur.