

# La Guía de Azure para desarrolladores

*Segunda edición*



PUBLICADO POR  
Microsoft Press  
Una división de Microsoft Corporation  
One Microsoft Way  
Redmond, Washington 98052-6399

Copyright © 2018 de Microsoft Corporation

Todos los derechos reservados. Ninguna parte del contenido de este libro puede reproducirse ni transmitirse en forma alguna ni por ningún medio sin la autorización previa por escrito de la editorial.

Los libros de Microsoft Press están a la venta en librerías y distribuidores de todo el mundo. Si tiene alguna duda relativa a este libro, escriba un correo electrónico a Microsoft Press Support, a la dirección [mssinput@microsoft.com](mailto:mssinput@microsoft.com). Denos su opinión acerca de este libro en <http://aka.ms/tellpress>.

Este libro se proporciona "tal cual" y expresa los puntos de vista y opiniones de los autores. Los puntos de vista, las opiniones y la información vertidos en este libro, incluidas las direcciones URL y otras referencias a sitios web de Internet, pueden cambiar sin previo aviso.

Algunos ejemplos recogidos en este libro tienen un carácter únicamente ilustrativo y son ficticios. No debe suponerse ni derivarse ninguna asociación o conexión real.

Microsoft y las marcas comerciales enumeradas en <http://www.microsoft.com>, en la página web de marcas comerciales, son marcas comerciales del grupo empresarial Microsoft. Todas las demás marcas son propiedad de sus respectivos titulares.

# Contenido

<b>Capítulo 1: La Guía de Azure para desarrolladores .....</b>	<b>1</b>
Introducción.....	2
La tecnología de la información como utilidad, no como carga .....	2
La computación como utilidad .....	2
Tipos de implementación de computación en el cloud .....	4
El cloud público .....	4
El cloud privado .....	4
El cloud híbrido .....	4
Descripción de los tipos de computación en el cloud .....	5
Infraestructura como servicio.....	5
Plataforma como servicio .....	6
Función como servicio .....	6
Software como servicio.....	6
¿Por qué usar Azure?.....	7
Azure tiene alcance global .....	8
Azure es extremadamente resistente.....	9
Azure es compatible con casi todos los sectores.....	9
Azure se centra en las capacidades para los desarrolladores .....	10
Azure está abierto y es compatible con los marcos de trabajo de tu elección.....	10
Puedes supervisar tus servicios de Azure sobre la marcha .....	10
Estamos aquí para ayudarte .....	10
<b>Capítulo 2: Introducción a Azure.....</b>	<b>11</b>
Usa tus propios entornos de desarrollo integrado y editores.....	11
Usa la interfaz de línea de comandos de Azure para crear secuencias de comandos.....	11
Ejecuta lo que desees en Azure .....	12
¿Y los costes? .....	12
Seleccionar los servicios adecuados de Azure .....	12

Dónde debes hospedar tu aplicación .....	12
¿Qué se debe usar en cada caso? .....	18
Crear aplicaciones más rápidas .....	19
Azure Content Delivery Network .....	19
Azure Redis Cache .....	19
Azure Traffic Manager .....	20
Dónde almacenar los datos .....	21
<b>Capítulo 3: Agregar inteligencia a tu aplicación .....</b>	<b>26</b>
Búsqueda de Azure .....	26
Azure Cognitive Services .....	27
Azure Bot Service .....	27
Usar eventos y mensajes en una aplicación .....	28
Azure Service Bus .....	28
Azure Event Hubs .....	29
Azure IoT Hub .....	29
Azure Event Grid .....	30
¿Qué se debe usar en cada caso? .....	31
<b>Capítulo 4: Proteger una aplicación .....</b>	<b>32</b>
Azure Active Directory .....	32
Azure API Management .....	32
Azure Key Vault .....	33
Protección de Azure contra ataques de denegación de servicio distribuidos .....	33
Firewall de aplicaciones web de Azure .....	33
Cifrado predeterminado de datos .....	34
<b>Capítulo 5: Dónde y cómo implementar los servicios de Azure .....</b>	<b>35</b>
El punto de vista de DevOps .....	35
Plantillas de Azure Resource Manager .....	36
Azure Service Fabric .....	36
Contenedores de Azure .....	37
Azure Stack .....	38
Dónde y cuándo implementar .....	38
<b>Capítulo 6: Tutorial sobre Azure .....</b>	<b>39</b>
Tutorial n.º 1: Azure Portal .....	39
Mosaicos en Azure Portal .....	39
Crear una nueva máquina virtual .....	41
Tutorial n. 2: desarrollar una aplicación web de Node.js con MongoDB en Azure .....	45
Crear la aplicación web y la base de datos con Azure Portal .....	45
Tutorial n. 3: crear un back-end para una aplicación móvil con Azure .....	54

Crear una aplicación móvil a través de Azure Portal .....	54
Características adicionales y paso a la producción .....	58
<b>Capítulo 7: Usar Azure Marketplace .....</b>	<b>59</b>
¿Cómo puede ayudarme Azure Marketplace como desarrollador? .....	59
Soluciones de Azure.....	60
<b>Resumen y próximos pasos.....</b>	<b>60</b>
<b>Sigue aprendiendo con una cuenta gratuita de Azure .....</b>	<b>60</b>
<b>Acerca de los autores .....</b>	<b>61</b>

# La Guía de Azure para desarrolladores

La demanda de software y servicios relacionados es hoy en día mayor que nunca, y esta tendencia está llamada a continuar de un modo exponencial a medida que aplicaciones de todo tipo pasen a formar parte de todos los aspectos de nuestra vida cotidiana. Tú, como desarrollador, eres el “mago” que está llevando esta revolución digital a tus usuarios, y eso no es fácil. Además de crear las características tangibles únicas que ofrece el software, debes tener en cuenta un gran número de factores: tus aplicaciones deben tener un alto grado de disponibilidad y rendimiento en todo el mundo y deben funcionar sin problemas en una amplia gama de dispositivos, ser seguras y poder ejecutarse a gran escala por un precio razonable. Estos desafíos son comunes a la mayor parte del software y la mayoría de los sectores. Las soluciones, sin embargo, son algo efímero que reinventamos una y otra vez, y crearlas y mantenerlas puede resultar difícil y costoso.

Hemos creado la *Guía de Azure para desarrolladores* para ayudarte en tu viaje al cloud, tanto si solo estás valorando la posibilidad de dar el paso como si ya lo has decidido y has iniciado el proceso. Este es un libro electrónico escrito por desarrolladores y para desarrolladores. Y pretende proporcionar a los desarrolladores un conocimiento básico sobre Azure y mostrarles lo que la plataforma les ofrece a ellos y a sus organizaciones y la manera de aprovechar todas estas ventajas.

En la segunda mitad de la guía se ofrecen escenarios como un recorrido por Azure Portal y la creación de una máquina virtual. También se analiza el desarrollo y la implementación de una aplicación web que utiliza Node.js y MongoDB. Se describen tareas típicas como la integración continua y la implementación continua (CI/CD), los entornos de ensayo, las escalas, el registro y la supervisión. Para finalizar, crearemos un back-end de una aplicación móvil que incluye autenticación y sincronización sin conexión.

# Introducción

Hay innumerables libros, artículos, documentos, publicaciones de blogs, etc., que describen las virtudes de la computación en el cloud, y puedes encontrar zetabytes de información que explican el concepto de computación en el cloud desde un punto de vista técnico. Pero antes de entrar en eso, debes plantearte un par de preguntas sencillas: ¿por qué debo cambiar al cloud? ¿Cuáles serán sus beneficios para mí, para mi equipo y para mi compañía?

## La tecnología de la información como utilidad, no como carga

La computación en el cloud es la computación como utilidad. ¿Qué quiere decir esto? Para ayudarte a visualizarlo, piensa en otra cosa omnipresente en tu vida: la electricidad.

¿Quién pensaría hoy en día que tener una miniplanta generadora fuera de cada casa y cada negocio sería un modelo eficiente para la distribución de energía eléctrica? Estamos acostumbrados a pulsar un interruptor para encender la luz y hacer lo que necesitamos; ni se nos ocurriría pensar en tener que comprar, instalar y mantener todo el equipamiento necesario para iluminar nuestro hogar o nuestra oficina. En lugar de eso, pagamos a la compañía eléctrica local por la potencia que consumimos cada mes y no tenemos que preocuparnos por la física subyacente a generación de electricidad, por el gasto de capital, por el mantenimiento del equipo, por la reparación después de las tormentas ni por nada. Pulsamos el interruptor y nos dedicamos a otra cosa.

Pues igual que ni necesitas ni quieres saber exactamente cómo llega la electricidad a tu casa, tampoco necesitas saber exactamente cómo se te suministran la capacidad y los servicios de computación que consumes en el cloud. Otros lo hacen por ti. Por tanto, trasladar la computación, el almacenamiento y la red al cloud proporciona, ante todo, facilidad de uso, como encender una lámpara. Y solo tienes que pagar por lo que usas, del mismo modo que solo pagas por la electricidad que consumes en tu hogar.

Por supuesto, si vamos más allá de esta analogía, el cloud ofrece muchos más beneficios a tus operaciones de TI, como la escalabilidad y la capacidad de utilizar servicios inteligentes como las búsquedas y el reconocimiento facial en tus aplicaciones, por mencionar solo dos.

## La computación como utilidad

Por tanto, en última instancia, puedes ver los servicios en el cloud, como Azure, como un ejemplo de *computación como utilidad*. Por supuesto, Azure ofrece muchos más productos y servicios que la compañía eléctrica local, así que la computación en el cloud puede ser a veces difícil de definir con claridad. De hecho, hay quien dice que el cloud consiste básicamente en usar *los equipos de otras personas*, lo cual es técnicamente cierto, pero el cloud es mucho más:



*El cloud es un sistema que proporciona **servicios inteligentes, sólidos y resistentes** y recursos de **computación** a una **escala infinita, elástica y global**.*

Vamos a analizar esta descripción:

El cloud y Azure proporcionan servicios que ayudan a lograr muchas cosas: desde lo cotidiano, como la adición de búsquedas a tu aplicación, a funciones más exóticas, como la implementación de flujos de trabajo de integración continua (CI) e implementación continua (CD). Puedes ajustar automáticamente tu base de datos o configurar notificaciones push para los dispositivos móviles de un modo rápido y sencillo. Estos son solo algunos ejemplos de cosas comunes que los desarrolladores han creado una y otra vez para su propio uso, pero que ahora están disponibles como servicios. Eso te permite utilizar estos servicios con muy poco esfuerzo, casi como si pulsases un interruptor de luz. Así puedes concentrarte en los aspectos de tu aplicación que la hacen única; las características que aportan un valor añadido real a tus usuarios.

Además de servicios, el cloud ofrece recursos de computación en forma de máquinas virtuales (VM), contenedores, bases de datos, etc. Puedes usarlos para hospedar aplicaciones o para proporcionar una infraestructura completa a tus usuarios.

Como ya se ha indicado, el poder del cloud es que los servicios y recursos son increíblemente sólidos y resistentes. Es muy poco probable que no funcionen. Esto se debe a que el cloud es inteligente. Tiene capacidad para repararse automáticamente y, al igual que en el caso de Azure, hay centros de datos en todo el mundo, equipados con decenas de miles de servidores. Si un servidor deja de funcionar, otro toma el control. Si un centro de datos completo deja de funcionar (algo muy poco probable), otro lo sustituye. Todo esto es posible gracias a la enorme escala del cloud.

Uno de los argumentos más convincentes a favor del cloud es que sus servicios y recursos se pueden ampliar de forma casi ilimitada, y desde luego hasta un grado que es imposible alcanzar con recursos on-premises, a menos que se esté dispuesto a gastar enormes sumas de dinero en equipamiento y personal para administrarlo todo. Y puedes aumentar su escala de forma global. Puedes ubicar tus servicios en cualquier lugar del mundo para proporcionar una experiencia de alto rendimiento a los usuarios estén donde estén. Esto también significa que puedes guardar los datos donde más te convenga.

Y algo aún más importante: al utilizar recursos del cloud, puedes volver a reducir la escala de tus servicios y recursos cuando disminuya la demanda. Volviendo otra vez a la analogía de la electricidad, si esperas muchos amigos y familiares en casa para una fiesta, encenderás muchas luces dentro y fuera de la casa, usarás el microondas con frecuencia, pondrás música y, en general, consumirás más electricidad. Pero cuando la fiesta termine y los invitados se marchen, el consumo de energía volverá a ser el normal, al igual que tu factura. Lo mismo ocurre cuando se utilizan el cloud y Azure: solo se paga por lo que se utiliza, no por lo que tal vez se necesite cuando el uso aumente.



**Más información** Para obtener información sobre el portal de Azure y crear tu primera máquina virtual, visita <https://docs.microsoft.com/azure/virtual-machines/windows/quick-create-portal>.

Además de una gran escalabilidad, servicios inteligentes listos para usar y la eficiencia del pago por uso, el cloud ofrece mayor seguridad.

Millones de personas usan el cloud ininterrumpidamente en todo el mundo, y por supuesto son también muchas las personas que lo atacan. Proveedores de prestigio y con experiencia en el cloud como Microsoft conocen los patrones de uso de los usuarios normales y los de las personas malintencionadas. Esto quiere decir que saben cómo proteger frente a los ataques, tanto los más frecuentes como los más inusuales. Las herramientas de supervisión inteligentes, los algoritmos de Machine Learning y la inteligencia artificial permiten a los proveedores del cloud detectar los ataques en tiempo real y detenerlos en seco.

Las [décadas de experiencia en seguridad](#) y el tráfico de gran escala, combinados con una excepcional experiencia en la seguridad del sector, convierten el cloud en un entorno mucho más seguro que cualquier centro de datos on-premises.



**Más información** Para obtener más información sobre los procedimientos empleados por Azure para proteger tus aplicaciones y tus datos, consulta [Cómo detecta Azure Security Center los ataques distribuidos de denegación de servicio \(DDoS\) por medio de la inteligencia sobre ciberamenazas, Información general sobre seguridad de Azure y Cómo actúa Microsoft ante los delitos informáticos](#).

Hemos explorado brevemente los motivos por los que quieres iniciar la migración al cloud y a Azure. Ahora, vamos a examinar los componentes principales del cloud, lo que hacen y cómo encajan en el esquema general.



# Tipos de implementación de computación en el cloud

El cloud ofrece varios niveles de productos y servicios, tales como infraestructura como servicio (IaaS) y plataforma como servicio (PaaS), pero antes de hablar de ellos vamos a dar un paso atrás para centrarnos en los tres tipos básicos de plataformas en el cloud: *pública*, *privada* e *híbrida*. Estos términos se han convertido rápidamente en palabras de moda en el ámbito de la tecnología actual. Vamos a analizarlos.

## El cloud público

En el modelo de cloud público, el usuario no posee ningún hardware; el proveedor del cloud es el responsable de adquirir y mantener todo el hardware. El cloud público está integrado por todos los servicios y recursos de computación que el usuario no posee, sino que utiliza a través de su proveedor de cloud. El proveedor es responsable de que todo funcione y de respetar los acuerdos de nivel de servicio (SLA). *El usuario paga por lo que usa, no por lo que posee.*

## El cloud privado

Con un cloud privado, el usuario posee todo el hardware o, al menos, tiene pleno control sobre él. El hardware que ejecuta tus servicios y hospeda tus datos está en algún lugar de tu centro de datos on-premises. Por supuesto, este control pleno tiene un precio: debes comprarlo y mantenerlo todo. *Pagas por lo que posees, no por lo que usas.*

Pero ¿cómo se configura una plataforma de cloud privado en un centro de datos on-premises tradicional? Afortunadamente, Azure proporciona una solución única diseñada específicamente para funcionar en este entorno: Azure Stack. En pocas palabras, Azure Stack es tu propia instancia privada de Azure, que puedes ejecutar en tu centro de datos on-premises.

Ejecutar servicios en un cloud privado no es lo mismo que ejecutar servicios en un centro de datos on-premises. [Azure Stack](#) ofrece todas las ventajas de Azure, incluidos los servicios inteligentes que puedes usar en tus aplicaciones, todo dentro de los límites de tu centro de datos on-premises. Así, aunque por razones especiales de gobernabilidad o de seguridad necesites mantener ciertas operaciones “cerca de casa” en tu centro de datos on-premises, podrás disfrutar de todos los beneficios ofrecidos por Azure y el modelo del cloud.

## El cloud híbrido

El modelo de cloud híbrido es, como su nombre indica, una mezcla del cloud público y el privado. Puedes utilizar servicios de cloud público que usen recursos de tu cloud privado y viceversa. En Azure, podrías ejecutar una aplicación en Aplicación web de Azure que se conecte a una base de datos on-premises utilizando conexiones híbridas de Azure. Esto abre muchas posibilidades. Te permite controlar dónde están tus aplicaciones y tus datos y disfrutar a la vez de los beneficios de usar servicios inteligentes en el cloud.

Azure ofrece soluciones para todo los tipos de implementaciones de computación en el cloud. Esto da a las compañías más libertad para usar Azure donde y como deseen. Y pueden determinar el grado de control que quieren tener sobre sus aplicaciones y sus datos.

# Descripción de los tipos de computación en el cloud

En la computación en el cloud, podemos clasificar los servicios y los recursos por tipos. Hay muchas definiciones diferentes de los tipos de computación en el cloud. Los tipos se diferencian por el nivel de control y las responsabilidades que el usuario tiene frente a la cantidad de tiempo y esfuerzo que puede dedicar a aumentar el valor que ofrece a su compañía y a sus clientes. La figura 1-1 y la figura 1-2 ilustran cómo se definen los tipos de computación en el cloud para Azure.

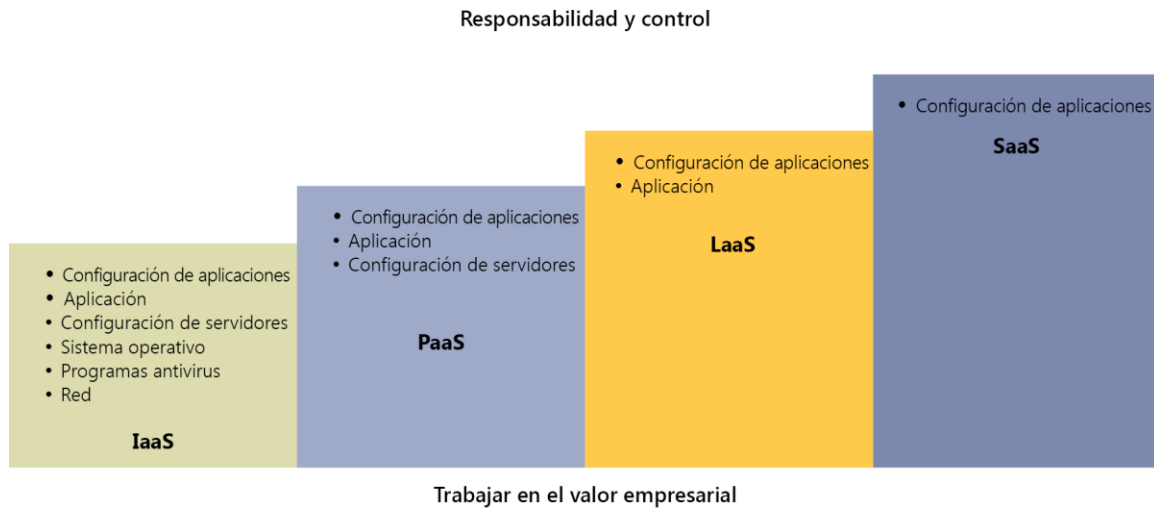


Figura 1-1: tipos de computación en el cloud

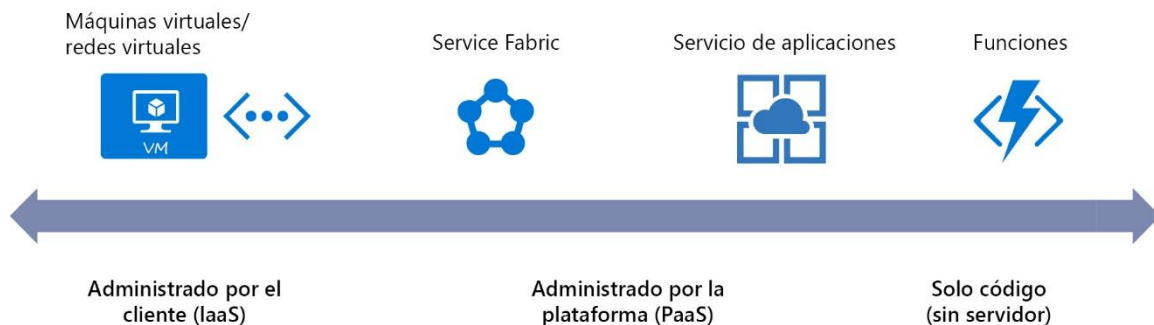


Figura 1-2: ejemplos de ofertas basadas en el cloud

## Infraestructura como servicio

Con la infraestructura como servicio (IaaS [pronunciado "ai as"]), tú eres responsable de todo, *excepto* el hardware. Azure se encarga de todo el hardware, como las unidades de estado sólido (SSD), el cableado de red, los enrutadores, las fuentes de alimentación, los generadores de reserva, los sistemas de refrigeración, etc. Los servicios incluidos en esta categoría son, entre otros, [Azure Virtual Machines](#), [Azure Virtual Network](#) y [contenedores de Azure](#).

Tú te encargas de todo lo demás. Tu aplicación, las revisiones del sistema operativo (SO), la configuración de la red lógica e incluso de mantener actualizados los programas antivirus en los equipos. Esto significa que conservas un control total sobre estos elementos. Pero también significa que dedicas mucho tiempo y esfuerzo a mantener en funcionamiento los servicios y los recursos, y menos tiempo a trabajar para añadir valor de negocio a tus principales productos de aplicaciones.

## Plataforma como servicio

En el modelo de plataforma como servicio (PaaS [pronunciado “pas”]), tú eres responsable de tu aplicación y de la configuración de esta; Azure proporciona el sistema operativo y el hardware. Como se muestra en la figura 1-1, PaaS es un nivel de abstracción superior al de IaaS. El usuario tiene muchas menos responsabilidades, pero menos control directo sobre los recursos de hardware que con el modelo IaaS. Los servicios incluidos en la categoría de PaaS son [Azure App Service](#), [Azure SQL Databases](#), [Azure Redis Cache](#) y [Azure Batch](#).

Con PaaS, puedes hospedar la aplicación, pero este nivel también ofrece capacidades adicionales que puedes utilizar “out of the box”, como las notificaciones push para dispositivos móviles mediante Aplicación móvil de Azure y el ajuste automático de la base de datos en Azure SQL Database.

Puedes implementar la aplicación y configurarla. También puedes llevar a cabo la configuración básica del servidor, como ajustar la escala del nivel de precios o el número de instancias en las que se ejecuta tu aplicación. La aplicación se sigue ejecutando en un servidor que está en otra ubicación, pero tú no eres responsable de ese servidor (ni tienes ningún control sobre él), lo que significa que puedes dedicar mucho más tiempo a agregar valor de negocio a las aplicaciones y menos tiempo a mantener un servidor en funcionamiento.



**Pruébalo** Crea una aplicación de Node.js con MongoDB en Azure en este tutorial en <https://docs.microsoft.com/azure/app-service/app-service-web-tutorial-nodejs-mongodb-app>.

## Función como servicio

En el modelo de función como servicio (FaaS [pronunciado “fas”]), solo eres responsable de tu aplicación y de tu lógica de negocio. No necesitas preocuparte por la escala, el sistema operativo o el hardware. El modelo FaaS se denomina a veces *computación sin servidor*. Los servicios de la categoría de FaaS son [Azure Functions](#), [Azure Logic Apps](#) y [Azure Event Grid](#).

En el modelo FaaS, solo tienes que crear tu aplicación o aplicar la lógica de negocio y ejecutarla. El ajuste de la escala se produce de forma automática. Uno de los beneficios más destacables del modelo FaaS es que solo se paga por los recursos que se usan cuando se ejecuta la lógica en lugar de pagar por un servicio que está siempre activo, esperando a que alguien lo utilice.

Pero FaaS no se limita a hospedar tu aplicación; conecta tu lógica a los desencadenadores y las fuentes. Las fuentes externas, como las colas de almacenamiento de Azure o WebHooks, pueden iniciar Azure Functions o Logic Apps y proporcionarles datos para procesar. No tienes que escribir la lógica de activación ni configurar las conexiones con los almacenes de datos o servicios externos, lo que significa que puedes centrarte en tu lógica y en las características destinadas al cliente.

## Software como servicio

Con el software como servicio (SaaS [pronunciado “sas”]), solo tienes que preocuparte de configurar el software, nada más. De hecho, SaaS es el nivel de abstracción más alto de los tipos de computación en el cloud. Los servicios de la categoría SaaS son, entre otros, [Azure Cognitive Services](#) y [Azure IoT Suite](#). [Office 365](#) es otro ejemplo de un conjunto de aplicaciones SaaS de Azure y ofrece aplicaciones comerciales fiables como Microsoft Word, Excel, PowerPoint y otras. Y todo esto está disponible sin tener que instalar o mantener nada.

Con el modelo SaaS, el software está listo para usar. Todo lo que tienes que hacer es configurar tus preferencias y empezar a trabajar. No tienes que escribir el software ni preocuparte por la implementación, el escalado, los sistemas operativos y el hardware.

## ¿Por qué usar Azure?

Crees que el cloud podría ser lo que necesitas para que tu compañía cree productos de software increíbles y ofrezca el valor y la experiencia de usuario que tus clientes esperan. Sin duda, la plataforma en el cloud Azure puede ayudarte a lograrlo. Con sus numerosos recursos y su potencia de computación, además de una amplia gama de servicios inteligentes, puedes crear aplicaciones dinámicas con un alto grado de escalabilidad y pagar solo por lo que usas. Pero hay muchos proveedores de cloud hoy en día y cada uno ofrece capacidades y productos únicos. ¿Por qué tendrías que optar por Azure?

Con Azure, contarás con el profundo conocimiento y la experiencia de Microsoft, una compañía que lleva más de cuarenta años a la vanguardia de las soluciones de computación y TI personales y empresariales. Microsoft lleva décadas diseñando, creando y ejecutando enormes redes y operaciones de centros de datos para sus propios fines. Toda esa experiencia y esos conocimientos forman parte ahora del diseño de Azure. Constituyen su base.

### ¿Quién usa Azure?

[Adobe](#), [Jet.com](#), [Geico](#), [GeekWire](#) y [DocuSign](#) son solo algunos ejemplos de clientes de todo el mundo que usan la plataforma Azure para alcanzar sus objetivos de negocio y generar más valor para sus clientes. Entre esos objetivos se incluyen la solicitud de App Services, computación sin servidor, inteligencia artificial, Azure IoT e incluso streaming multimedia. Veamos algunas usos interesantes que los clientes han dado a la plataforma Azure para satisfacer sus necesidades de negocio.

El gigante industrial surcoreano [Samsung](#) crea todo tipo de productos: desde teléfonos a televisores, lavadoras, grandes electrodomésticos y aparatos de aire acondicionado. La compañía decidió utilizar Azure para aumentar la eficiencia y la solidez de sus aparatos de aire acondicionado mediante un servicio de supervisión y mantenimiento remotos. Samsung utiliza Azure IoT Hub para recopilar grandes cantidades de datos enviados desde sensores integrados en sus equipos de aire acondicionado que supervisan el estado de las máquinas y el entorno en el que se ejecutan. Estos datos se capturan en Azure SQL para realizar análisis en tiempo real. Los ingenieros pueden utilizar la información que se extrae para ajustar el comportamiento de los equipos de aire acondicionado, lo que genera un ahorro de energía del 15 al 18 % en promedio para sus clientes.

La compañía [Coca-Cola](#), una de las marcas más conocidas del mundo, introdujo su bebida homónima en 1886 y ha sido líder en todo el planeta desde entonces. En 2012, la compañía lanzó Coca-Cola Journey, un proyecto y un sitio web muy ambiciosos cuyo propósito es ayudar a Coca-Cola a conectar mejor con una nueva generación de clientes que cada vez usan más los canales de marketing no tradicionales para decidir sus compras. La compañía quería aprovechar las redes sociales y un servicio de Microsoft llamado How-Old.net para crear una experiencia interactiva divertida y atractiva para los usuarios que le permitiera a la vez conocer mejor a sus clientes. How-Old.net, basado en la tecnología de Cognitive Services analiza datos extraídos de imágenes faciales para establecer la edad y el sexo de los usuarios. Pero Coca-Cola pidió al equipo de Microsoft que llevara el proyecto un paso más allá del reconocimiento facial mediante el uso de modelos de Machine Learning para detectar objetos en las imágenes. Durante un mes, el sitio invitó a los usuarios a enviar una foto suya con una botella de Coca-Cola con un logotipo visible para ver qué sucedía. Si la aplicación web detectaba una botella, aparecía la antigüedad de esta y la página se volvía roja. Y se animaba a los aficionados a la coca-cola a compartir su experiencia en las redes sociales y a seguir explorando el sitio de Coca-Cola Journey para saber más sobre la venerable botella curvilínea.

[CarMax](#), el mayor vendedor minorista de coches de segunda mano de Estados Unidos, rediseñó su sitio web utilizando los servicios de la plataforma Azure para responder con más rapidez a los 16 millones de personas que visitan su sitio cada mes. En lugar de “trasladar” el sitio a un entorno IaaS de Azure, CarMax decidió reconstruir el sitio con la oferta PaaS de Azure. De este modo, CarMax pudo modernizar el sitio con nuevas funcionalidades del cloud, hacerlo totalmente compatible con los dispositivos móviles y, sobre todo, avanzar hacia un modelo de desarrollo dinámico de DevOps. CarMax está avanzando rápidamente hacia una arquitectura de microservicios que permitirá a la compañía ampliar y actualizar partes de aplicaciones independientemente de las otras partes. Aunque esto tardará años en lograrse, elegir Azure como base de desarrollo ayudará al vendedor de automóviles a llegar a los microservicios más deprisa.

[NBC Sports](#) trabaja con Microsoft para transmitir los eventos con una difusión masiva. En 2016, las dos compañías colaboraron para transmitir los Juegos Olímpicos de Río de Janeiro. NBC usó Azure Media Services para transmitir contenido de más de 50 eventos olímpicos al mismo tiempo, con un perfil de 1080p y para la increíble cifra de 100 millones de usuarios únicos que consumieron 2710 millones de minutos de contenido desde una amplia gama de dispositivos y factores de forma. Media Services proporciona la escalabilidad y la solidez que se necesitan para suministrar este contenido sin problemas a los espectadores.

Azure permite a las compañías, grandes y pequeñas, y a organizaciones gubernamentales, instituciones educativas y a cualquier usuario de los servicios de TI administrar de una forma más sencilla y rentable sus mundos digitales, tanto para ellos mismos como para sus clientes. En síntesis, permite:

- **Ofrecer experiencias de software multiplataforma, por ejemplo experiencias móviles, de escritorio, web e híbrida:** Azure proporciona servicios con los que se pueden implementar estos escenarios “out of the box”, como Azure App Service, que incluye Aplicación web y Aplicación móvil y Mobile Center y Visual Studio Team Services para suministrar y administrar todo tipo de aplicaciones.
- **Suministrar servicios donde se necesitan:** Azure es único por su amplia variedad de opciones de implementación: el cloud público, soluciones híbridas y Azure on-premises, en su propio centro de datos (a través de Azure Stack). Todas estas opciones pueden proporcionar una enorme escala a un precio asequible, mientras tus datos se mantienen donde deben estar.
- **Deliver an intelligent, massive scale, data platform:** Azure has many services that you can use to capture, store, analyze, and present your data. Son servicios como Azure SQL Databases, Data Lake Store y Data Lake Analytics, HDInsight, Event Hubs, Cognitive Services y muchos más. Todos estos servicios pueden crear una plataforma de datos asequible, inteligente y con una gran capacidad de ampliación que proporciona a las compañías las herramientas que necesitan para extraer conocimientos valiosos de la enorme cantidad de datos que manejamos hoy en día.
- **Suministrar software de alta calidad con rapidez:** tienes que hacer pruebas siempre y detectar los errores con rapidez para suministrar software de calidad. Con Azure puedes hacerlo a través de mecanismos de CD y CI integrados directamente en servicios como Aplicación web o a través de Visual Studio Team Services. Características como las ranuras de implementación de App Services permiten entregar las aplicaciones con rapidez sin tiempos de inactividad. Y, para que puedas comprobar que tu aplicación está funcionando como se esperaba en la producción, Azure proporciona servicios de supervisión, como Application Insights, que permiten saber exactamente cómo funciona una aplicación y qué se puede mejorar.

## Azure tiene alcance global

Con centros de datos en más de 42 países y regiones de todo el mundo, la increíble cobertura de Azure ofrece numerosos beneficios. En primer lugar, puedes reducir cualquier tiempo de inactividad que las aplicaciones puedan experimentar si se produce un error de un centro de datos completo

(algo extremadamente improbable), porque la aplicación puede conmutar por error a otro centro de datos y seguir funcionando casi como si no hubiese pasado nada. Esto también puede reducir los costes de recuperación ante desastres.

Si utilizas redes de entrega de contenido (CDN) para reducir la latencia, es muy probable que haya un centro de datos de Azure cerca de tus clientes y de las ubicaciones en las que tienes que entregar su contenido. Otro beneficio de Azure con tantas instalaciones en tantos lugares es que cuenta con una gama más amplia de opciones para el almacenamiento de datos. Esto puede ayudarte a garantizar que tu compañía cumple los reglamentos y las leyes locales o regionales con respecto a la soberanía de los datos.

Microsoft sigue realizando grandes inversiones en infraestructura e innovación de centros de datos [por medio del desarrollo de hardware de código abierto](#) y nuevas soluciones de centros de datos como los *centros de datos submarinos* del [proyecto Natick](#).

## Azure es extremadamente resistente

Azure es una plataforma inteligente con supervisión y reparación automática en la que puedes confiar para mantener tus aplicaciones disponibles y en funcionamiento. Además, todos los servicios de Azure operan sujetos a [acuerdos de nivel de servicio \(SLA\) integrales](#) que definen su rendimiento en detalle. Los tiempos de actividad de los SLA van desde el 99,9 % (tres nueves) al 99,99 % (cuatro nueves). Esto significa que los servicios Azure son extremadamente resistentes y fiables. Si tus servicios no cumplen con las especificaciones del SLA suscrito, podrías optar a un descuento de hasta el 100 % de los costes de servicio en tu factura de Azure.

Para asegurarte de que los servicios siguen a pleno rendimiento, Microsoft supervisa el estado de cada servicio de Azure en todos los centros de datos de todo el mundo. Estas métricas están disponibles para que tú también puedas supervisarlas [en la página de estado pública de Azure](#).

Como ya se ha mencionado, Microsoft gestiona centros de datos y proporciona servicios de TI a gran escala desde 1989. A lo largo de ese tiempo ha acumulado muchos conocimientos. Hoy en día, Microsoft tiene algunas de las mejores mentes del sector de la TI trabajando continuamente para que Azure funcione sin interrupciones con el fin de que los usuarios puedan confiar en él para hospedar sus servicios y almacenar sus datos. Por eso, todos los servicios de Azure están sujetos a SLA que garantizan el funcionamiento.

## Azure es compatible con casi todos los sectores

[Azure ofrece el conjunto de ofertas de cumplimiento más completo de cualquier proveedor del cloud.](#)

La lista de cumplimientos es enorme y creciente continuamente. Entre los estándares que cumple Azure se incluyen [ISO 27001](#), [HIPAA](#) y [SOC 3](#). Entre los cumplimientos específicos de las distintas regiones se incluyen [EU-US Privacy Shield](#) y [China DJCP](#).

Por ejemplo, al cumplir con la certificación ISO 27001, Microsoft garantiza que implementa, supervisa, mantiene y mejora continuamente sus estándares de seguridad para su red global de centros de datos, así como para los distintos servicios de Azure.



**Más información** Para obtener una lista completa de las declaraciones de cumplimiento, visita <https://www.microsoft.com/trustcenter/compliance/complianceofferings>.

Azure tiene incluso una oferta llamada [Azure Government](#) destinada a los clientes gubernamentales de Estados Unidos, que es básicamente una "edición especial" del cloud de Azure que responde a las necesidades propias de las entidades gubernamentales. Azure Government se ejecuta fuera del cloud público de Azure, en su propia *instancia*, lo que significa que se ejecuta en servidores exclusivos en una infraestructura totalmente independiente y aislada.

## Azure se centra en las capacidades para los desarrolladores

Azure te ayuda, como desarrollador, a ser más productivo proporcionando gran parte de la infraestructura que, de lo contrario, tendrías que crear tú mismo. Con capacidades como el escalado automático y las herramientas de autenticación/autorización, puedes agregar características y funciones avanzadas y funcionalidad a tus aplicaciones con un esfuerzo mínimo o nulo. Puedes integrar servicios como Búsqueda de Azure o Cognitive Services que mejoren tus aplicaciones y respondan a las necesidades de los usuarios.

## Azure está abierto y es compatible con los marcos de trabajo de tu elección

Escribe tus aplicaciones en JavaScript e impleméntalas en Aplicación web. O escríbelas en tu entorno preferido, ya sea Ruby, Django, Java, PHP o .NET. Azure es compatible con muchos lenguajes. Puedes incorporar una aplicación ya existente o escribir una desde cero y se ejecutará en Azure.

Además, puedes elegir el ecosistema que prefieras. Azure es compatible con la mayoría de los sistemas operativos, como Linux y Windows, y puedes crear todo tipo de secuencias de comandos con Bash si lo deseas. También puedes ejecutar tu aplicación como desees: usando contenedores, Azure Service Fabric, App Services, Azure Stack, etc.

Una vez que hayas aprendido a usar Azure con un conjunto de herramientas, podrás usarlo con cualquier otro conjunto de herramientas. Los servicios y Azure Portal funcionan siempre del mismo modo.

## Puedes supervisar tus servicios de Azure sobre la marcha

¿Eres una de esas personas a las que les gusta tener controladas las operaciones todo el tiempo? Con Azure, puedes supervisar el rendimiento de tus servicios y solucionar los problemas que puedan surgir mediante el uso de la [aplicación móvil de Azure](#). Esta útil aplicación se ha diseñado para permitirte visualizar tus recursos incluso cuando no estés sentado en tu mesa en tu equipo principal. Puedes realizar tareas como reiniciar una aplicación web y detener una máquina virtual. También proporciona una completa experiencia de interfaz de la línea de comandos (CLI) a través de Cloud Shell, lo que te permite hacer prácticamente todo lo que se puede hacer con Azure Portal.

Además, hay servicios de Azure que realizan detecciones y recomendaciones inteligentes para tus servicios y pueden enviar alertas sobre ellos. Por ejemplo, pueden detectar que una aplicación web está funcionando más despacio de lo habitual o informarte de que un App Service se ha escalado automáticamente. Azure Security Center y Azure Monitor son algunos de los servicios que pueden alertarte sobre estas cosas para que nunca te pierdas nada importante.

## Estamos aquí para ayudarte

Hay muchas maneras de obtener ayuda con Azure si la necesitas.

Puedes comprar [un plan de soporte](#), que te da acceso a los equipos de soporte técnico de Azure y te proporciona otros servicios como orientación basada en prácticas recomendadas para el diseño para el cloud o ayuda para planificar la migración. En función de tus necesidades, puedes comprar un plan de soporte que garantice las respuestas de los equipos de soporte técnico en un plazo máximo de 15 minutos.

Además, puedes recibir ayuda gratis de forma ininterrumpida a través de muchos canales. Puedes enviar un tuit a [@AzureSupport](#) o contactar con los equipos de soporte a través de los [Foros de MSDN](#), [StackOverflow](#), [Reddit](#) o [Microsoft Tech Community](#).



# Introducción a Azure

Tras una profunda reflexión, te has unido a la comunidad de Azure y ahora quieres empezar a crear aplicaciones. ¿Qué necesitas? Realmente, no mucho. Lo más importante es contar con una conexión a Azure para implementar tu aplicación. Puedes utilizar las herramientas, las aplicaciones y los marcos de trabajo que prefieras.

## Usa tus propios entornos de desarrollo integrado y editores

Uses lo que uses para crear o editar tus aplicaciones en tu entorno on-premises, puedes utilizarlo con Azure. Puedes elegir lo que desees: desde el Bloc de notas a [Visual Studio Code](#), [Sublime Text](#), [Visual Studio](#) o los editores de código de Azure Portal. Tú decides.

Hay complementos disponibles para los principales IDE, lo que te permite hacer cosas como publicar directamente en Azure. Y si prefieres utilizar una herramienta sencilla como el Bloc de notas, puedes automatizar tus implementaciones en Azure usando las prácticas de integración continua (CI)/entrega continua (CD).

## Usa la interfaz de línea de comandos de Azure para crear secuencias de comandos

Azure proporciona una útil [interfaz de la línea de comandos \(CLI\)](#) con la que puedes hacerlo básicamente todo en Azure. Puedes utilizar la CLI de Azure para arrancar y detener máquinas virtuales (VM), para implementar tu aplicación en una app en Aplicación web de Azure, para crear nuevos recursos y para todo lo demás. También resulta muy útil para automatizar tareas y ejecutarlas en tus procesos de CI/CD.



Puedes usar la CLI de Azure disponible en Azure Portal en [Azure Cloud Shell](#) o localmente en tu máquina a través de las herramientas de la CLI de Azure. Otra ventaja es que puedes aplicar tus conocimientos de Bash o Windows PowerShell en la CLI de Azure.

## Ejecuta lo que desees en Azure

Azure es ideal para aplicaciones web y API. También es excelente como host para aplicaciones de escritorio y móviles. Por ejemplo, podrías utilizarlo para autenticar a los usuarios en tu aplicación de escritorio o enviar notificaciones push a tu aplicación móvil. Sea cual sea el escenario que tienes en mente, Azure puede agregarle valor.

### ¿Y los costes?

Sin duda, es importante saber lo que cuestan las cosas en Azure. Para ayudarte con eso, Microsoft ofrece la [calculadora de precios de Azure](#). Con esta práctica herramienta, puedes seleccionar los servicios que vas a utilizar y especificar cuánto prevés usarlos, y la calculadora te permitirá saber cuál será tu factura mensual y anual. Esto, junto con la cantidad de datos que uses, determinará tus costes.



**Nota** La calculadora de precios de Azure estima los costes para las suscripciones de pago por uso. Hay otras opciones de pago para empresas, partners y suscriptores de MSDN que pueden afectar los costes de los servicios.

También puedes supervisar tus costes con las [API de facturación de Azure](#) y con [alertas de precios](#). Puedes usar estas opciones para conocer los costes y las tendencias de gasto con el objetivo de controlar tus costes.

## Seleccionar los servicios adecuados de Azure

Azure proporciona una gran cantidad de servicios para ayudarte a crear y ejecutar tus aplicaciones. De hecho, hay tantos servicios que a veces puede resultar un poco confuso elegir los más adecuados para cada caso. ¿Cuáles debes elegir como desarrollador? Vamos a explorar algunas de las opciones.

### Dónde debes hospedar tu aplicación

La primera decisión que tendrás que tomar es dónde debes hospedar tu aplicación. Azure ofrece varias opciones de hosting.

#### Máquinas virtuales (VM)

Una de las formas de hospedar una aplicación es usar una máquina virtual en [Azure Virtual Machines](#). Esto te da mucho control sobre el modo de hospedar la aplicación, pero también te obliga a mantener el entorno, por ejemplo, a aplicar revisiones al sistema operativo (SO) y mantener actualizados los programas antivirus.

Puedes, por ejemplo, usar una máquina virtual para probar la última versión de preview de Visual Studio sin "ensuciar" tu equipo.

#### Azure Batch

Si necesitas ejecutar aplicaciones por lotes a gran escala o de computación de alto rendimiento (HPC) en las máquinas virtuales, puedes elegir [Azure Batch](#). Batch crea y administra una colección de hasta miles de máquinas virtuales, instala las aplicaciones que deseas ejecutar y programa trabajos en dichas máquinas. No necesitas implementar y administrar las distintas máquinas virtuales o clústeres

de servidores. Batch programa, administra y escala automáticamente tus trabajos, por lo que solo usas las máquinas virtuales que necesitas. Batch es un servicio gratuito; solo se paga por los recursos subyacentes consumidos, como las máquinas virtuales, el almacenamiento de información y la conexión en red.

Batch es un servicio idóneo para ejecutar cargas de trabajo paralelas a escala como modelos de riesgo financiero, transcodificación de contenido multimedia, VFX y representación de imágenes 3D, simulaciones de ingeniería y muchas otras aplicaciones que hagan un uso intensivo de los recursos informáticos. Utiliza batch para escalar una aplicación o un script que ya hayas ejecutado en estaciones de trabajo o en un clúster local, o para desarrollar soluciones SaaS que utilicen batch como plataforma de computación.



**Pruébalo** Empieza a usar Azure Batch en cinco minutos con [estos tutoriales paso a paso.](#)

## Contenedores

Los contenedores son mucho más ligeros que las máquinas virtuales y se pueden iniciar y detener en cuestión de segundos. Los contenedores ofrecen además una gran portabilidad que los hace idóneos para desarrollar una aplicación localmente en tu equipo y hospedarla después en el cloud, primero en la fase de prueba y más adelante en la producción. Puedes incluso ejecutar los contenedores on-premises o en otros clouds. El entorno que usas en su equipo de desarrollo viaja con el contenedor, por lo que tu aplicación se ejecuta siempre en el mismo ecosistema.

Al igual que las máquinas virtuales, los contenedores te proporcionan mucho control sobre tu entorno. Puedes instalar lo que necesites para ejecutar tus aplicaciones. Pero, de nuevo, serás el responsable de aplicar las revisiones y mantener el SO que se ejecuta en el contenedor, así como de otros elementos auxiliares como los programas antivirus.

## Hosting de contenedores con Azure Container Instances

Puedes hospedar contenedores usando [Azure Container Instances](#). Con este servicio, puedes integrar fácilmente un contenedor sin necesidad de un orquestador de contenedores, como Kubernetes, y sin tener que administrar los recursos que hospedan el contenedor.

El servicio Container Instances se factura por segundo, por CPU virtual, por gigabyte o por memoria.



**Más información** Para obtener más información sobre Container Instances, consulta [Introducción sobre Azure Container Instances](#).

## Hosting de contenedores con Azure Container Service

Otra forma de hospedar contenedores es usar [Azure Container Service](#). Con este servicio, puedes escalar y administrar los contenedores usando orquestadores como Mesosphere DC/OS, Docker Swarm o Kubernetes. Este servicio es una forma excelente para empezar a mover los contenedores al cloud.



**Más información** Para obtener más información sobre Container Service, consulta [Introducción sobre Azure Container Service](#).

## Azure Functions

Con Azure Functions, solo tienes que escribir el código que necesitas para una solución sin preocuparte por crear una aplicación completa o la infraestructura para ejecutarla. Una función es una unidad de lógica de código desencadenada por una solicitud HTTP, un evento de otro servicio de Azure o una programación. Los enlaces de entrada y de salida conectan el código de la función con

otros servicios, como Azure Blob Storage, Azure Cosmos DB y Azure Service Bus, con una cantidad mínima de código. Con Functions, puedes crear funciones con rapidez y hospedarlas en un entorno elástico que realiza el escalado de forma automática.

Otro aspecto que hace especial a Azure Functions es que puedes optar por pagar únicamente por las funciones que se ejecuten sin tener que mantener las instancias de computación en ejecución todo el mes. Esto se denomina también *sin servidor* porque solo tienes que crear tu aplicación sin preocuparte por los servidores ni por el escalado de estos.

Puedes escribir funciones de Azure en C#, F#, Node.js, Java, PHP y en una lista creciente de lenguajes.

Un ejemplo de una aplicación que usa Functions es una que activa una función cada vez que se carga un nuevo archivo de imagen en Azure Blob Storage. Al hacerlo, la función ajustaría el tamaño de la imagen y la escribiría en otra cuenta de Blob Storage. La firma de la función de este ejemplo tendría un aspecto similar al siguiente (en una secuencia de comandos de C#):

```
public static void Run(Stream myBlob, string name, TraceWriter log, Blob outputBlob) { }
```

Los datos de Blob que desencadenaron la función se pasan a la función como el parámetro `myBlob`, que incluye la dirección URL del Blob. Puedes usar el parámetro de enlace de salida `outputBlob` para especificar el Blob en el que deseas escribir el resultado. No es necesario escribir el código para conectar a Blob Storage, solo configurarlo.



**Pruébalo** [Crea tu primera función de Azure usando Azure Portal.](#)

## Azure Logic Apps

Puedes organizar la lógica de negocio con [Logic Apps](#), automatizar un proceso de negocio o integrar con aplicaciones de software como servicio (SaaS). Al igual que en el caso de Azure Functions, Logic Apps se puede activar desde una fuente externa; por ejemplo, un nuevo mensaje en una cola de Azure Storage. Puedes enlazar llamadas de la API a conectores para crear un flujo de trabajo (posiblemente complejo) en el que participen recursos del cloud y on-premises. Logic Apps tiene disponibles [muchos conectores con las API](#), como el que permite conectar con Azure SQL Databases, Salesforce, SAP, etc. También puedes exponer tus propias API o funciones de Azure como conectores para usarlos en una aplicación lógica, lo que te permitirá realizar fácilmente acciones en sistemas externos en tu flujo de trabajo o hacer que uno de ellos active tu aplicación lógica.

Al igual que las funciones de Azure, las aplicaciones lógicas no tienen servidor, se escalan automáticamente y solo se paga por ellas cuando se ejecutan.

El siguiente ejemplo muestra un flujo de trabajo de Logic Apps:

1. La aplicación lógica se activa mediante un correo electrónico que contiene un pedido de envío que llega a Office 365.
2. Usando los datos del correo electrónico, la aplicación lógica comprueba la disponibilidad del artículo pedido en SQL Server.
3. La aplicación lógica envía un mensaje de texto al teléfono del cliente utilizando Twilio (el número de teléfono estaba también en el correo electrónico), y en él se indica que el pedido se ha recibido y el artículo se ha enviado.



**Pruébalo** [Empieza a usar Azure Logic Apps.](#)

## Azure App Service

Como alternativa, puedes hospedar tus aplicaciones en una de las principales ofertas de servicio de Azure: [Azure App Service](#). Azure App Service es un conjunto de servicios de hosting y orquestación con características y capacidades comunes. Por ejemplo, todos los App Services tienen la capacidad de proteger una aplicación utilizando [Azure Active Directory](#) y pueden usar dominios personalizados.

Azure App Service consta de los siguientes elementos:

- **Aplicación web:** [Aplicación web](#) es uno de los servicios de Azure más usados. Puedes utilizarlo para hospedar tus aplicaciones web o tus API. Una aplicación web es básicamente una abstracción de un servidor web, como Internet Information Services (IIS) o Tomcat, que se utiliza para hospedar las aplicaciones basadas en HTTP.

Aplicación web puede hospedar aplicaciones que están escritas en .NET, Node.js, PHP, Java o Python, y existen extensiones que se pueden usar para ejecutar otros lenguajes.



**Pruébalo** Descubre cómo crear una aplicación de Node.js y MongoDB de muestra en nuestra sección de ejemplos.

- **Web App for Containers:** [Web App for Containers](#) te ayudará a implementar y ejecutar fácilmente aplicaciones web con contenedores a escala. Obtén imágenes del contenedor de Docker Hub o de Azure Container Registry, y Web App for Containers implementará la aplicación con contenedores con tus dependencias preferidas en producción en cuestión de segundos. La plataforma se encarga automáticamente de las revisiones del SO, del aprovisionamiento de capacidad y del equilibrio de carga.



**Pruébalo** Te guiamos por la implementación de una [aplicación web](#) con un contenedor con formato de Docker en nuestra sección de ejemplos.

- **Aplicación móvil:** [Aplicación móvil](#) proporciona un back-end para tus aplicaciones móviles. Puedes hospedar en Aplicación móvil una API con la que se conectan las aplicaciones móviles a través del SDK cliente multiplataforma. Está disponible para iOS, Android, Windows y Xamarin para iOS y Android y Xamarin Forms. Aplicación móvil ofrece características únicas como la sincronización sin conexión y las notificaciones push que te ayudan a crear una experiencia móvil segura, moderna y de alto rendimiento.

Puedes escribir tu back-end de Aplicación móvil en .NET o Node.js.



**Pruébalo** Te mostramos el procedimiento para crear un back-end móvil en nuestra sección de ejemplos.

## Características de Azure App Service

Azure App Service es uno de los principales servicios de Azure que se pueden usar para hospedar las aplicaciones. Cada uno de estos servicios ofrecen capacidades únicas, pero todos comparten algunas características comunes:

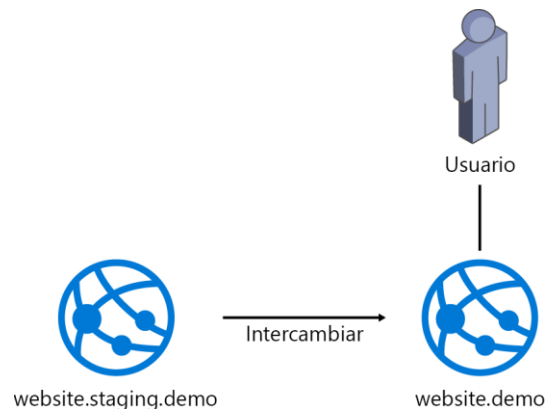
## Escalado

Azure App Service se ejecuta en [planes de App Service](#), que son abstracciones de máquinas virtuales. Una o más máquinas virtuales ejecutan Azure App Service, pero tú no necesitas saber cuáles porque Azure se encarga de ellas. Sin embargo, puedes ajustar la escala de los recursos que ejecutan Azure App Service. Puedes elegir un nivel de precios superior (desde gratis hasta premium) o aumentar el número de instancias de la aplicación que se ejecutan. Puedes incluso hacer que Azure App Service ajuste automáticamente el número de instancias en función de una programación o de métricas como la CPU, la memoria o la longitud de la cola HTTP.

## Ranuras de implementación

Esta es una característica muy útil de Azure App Service. Puedes implementar una nueva versión de tu aplicación en una ranura de implementación. En ella podrás probar si funciona del modo esperado y moverla después a la ranura de producción. Puedes incluso utilizar la característica *Pruebas en producción* [de Azure](#) para redirigir un porcentaje del tráfico de la aplicación de producción a una ranura de implementación. Por ejemplo, podrías desviar un 10 % de los usuarios a la nueva versión de tu aplicación en la ranura de implementación para ver si las nuevas características están funcionando como se esperaba y si los usuarios las están utilizando realmente.

Una vez que estés satisfecho con el funcionamiento de la nueva versión de tu aplicación en la ranura de implementación, puedes llevar a cabo un *intercambio*, que cambia la aplicación de la ranura de implementación por la de la ranura de producción. También puedes cambiar de una ranura de desarrollo a una ranura de ensayo y luego a la ranura de producción, como se ilustra en la figura 2-1. Antes de hacer esto, la operación de intercambio verifica que la nueva versión de tu sitio web se ha preparado y está lista para el uso. Cuando esto se confirma, la operación de intercambio alterna las ranuras y los usuarios ven la nueva versión de la aplicación sin tiempo de inactividad. Si lo deseas, también puedes revertir la implementación de la nueva versión.



**Figura 2-1:** intercambio de una ranura de implementación

Las ranuras de implementación se usan *dentro* de un entorno como desarrollo, prueba o producción. Las ranuras de implementación no se usan *como* entornos, porque residen en el mismo plan de App Service, y es preferible que los entornos estén separados por motivos de seguridad, escala, rendimiento y facturación.

Puedes intercambiar ranuras de implementación manualmente, a través de la interfaz de la línea de comandos (CLI) de Azure y a través de la API de administración de Azure. Esto permite que herramientas como Visual Studio Team Services realicen operaciones de intercambio durante un lanzamiento.

Una ranura de implementación es otro elemento de Azure App Service (como una aplicación web) que se ejecuta en el mismo plan de Azure App Service, junto con el Azure App Service original. Como las ranuras de implementación se ejecutan en el mismo plan de Azure App Service que el Azure App Service original, su uso no conlleva ningún coste adicional.

### *Entrega continua*

Para publicar tu aplicación en App Services, puedes usar servicios externos como Visual Studio Team Services, Jenkins u Octopus Deploy. También puedes usar la [característica de entrega continua \(CD\)](#) de App Services. Esto te permite crear una secuencia de compilación-prueba-lanzamiento en el propio App Service. El proceso hace lo siguiente:

1. Recupera el código fuente más reciente desde el repositorio indicado
2. Compila el código según la plantilla elegida (ASP.NET, Node.js, etc.)
3. Implementa la aplicación en un entorno de ensayo y realiza una prueba de carga
4. Implementa la aplicación en producción después de la aprobación (puedes indicar si deseas utilizar una ranura de implementación)

### *Conectar a recursos on-premises*

Se pueden conectar recursos externos como almacenes de datos a App Services. No es necesario que estos recursos se encuentren en Azure; pueden estar en cualquier sitio, como on-premises, en tu propio centro de datos. Puedes conectarte a servicios on-premises a través de muchos mecanismos, en función de tus necesidades. Puedes usar [Conexiones híbridas de Azure](#), [Azure Virtual Networks](#) y [Azure ExpressRoute](#) para conectarte a los recursos on-premises.

### *Dominios personalizados y certificados de Azure App Service*

Cuando se integra una aplicación en Azure App Service, expone una dirección URL, por ejemplo, <https://myazurewebsite.azurewebsites.net>. Es muy probable que desees usar tu propio dominio personalizado. Para ello, tienes que asignar ese nombre de dominio a App Services. [Las siguientes secciones describen cómo hacerlo](#).

Además, puedes asegurarte de que la aplicación se sirve a través de HTTPS usando un certificado de Capa de sockets seguros (SSL). Puedes traer tu propio certificado o comprar uno [directamente desde Azure Portal](#). Cuando se compra un certificado SSL desde Azure Portal, se compra un certificado de Azure App Service, que se puede configurar para el uso en los enlaces de dominios personalizados.



**Pruébalo** [Descubre cómo comprar y configurar un certificado en este tutorial](#).

### *Entorno de App Service*

En una aplicación web de varios niveles, a menudo se tiene una base de datos o servicios utilizados por una aplicación en Aplicación web. Lo deseable es que esos servicios solo estén expuestos a la aplicación y no a Internet. Por supuesto, la propia aplicación está a menudo expuesta a Internet porque proporciona el punto de acceso para los usuarios.

Para aislar estos servicios de soporte de Internet, puedes utilizar una Azure Virtual Network. Este servicio empaqueta tus servicios de soporte y los conecta a tu aplicación en Aplicación web de tal modo que los servicios de soporte solo están expuestos a la aplicación, no a Internet. [En este artículo](#) se describe este servicio con más detalle y se muestra cómo usarlo.

A veces, puedes desear tener aún más control. Tal vez desees que tu aplicación esté contenida en una red virtual para poder controlar el acceso a ella. Puedes desear que otra aplicación la invoque en Aplicación web y que forme parte de tu back-end. En ese caso, puedes utilizar un [Azure App Service Environment](#). Esto te proporciona una escala muy alta y te da control sobre el aislamiento y el acceso a la red. Sin embargo, debes tener en cuenta que App Service Environment solo está disponible para App Services en el nivel de precios premium.



**Nota** Actualmente, App Service Environment no funciona para Web App for Containers.

## ¿Qué se debe usar en cada caso?

Aunque App Service, Functions y Logic Apps suelen funcionar conjuntamente, cada uno de ellos se ha diseñado para responder a unas necesidades de aplicaciones concretas.

La tabla 2-1 describe brevemente la finalidad para la categoría de necesidades que puedes tener.

Tabla 2-1: Qué App Service se debe usar en cada caso

Categoría	Finalidad	Tipo de Azure App Service
Aplicaciones de host	Aplicaciones web de host y API	Aplicación web y/o Web App for Containers de App Service
	Back-end de host para aplicaciones móviles	Aplicación móvil de App Service
Organizar procesos	Organizar un paso en un proceso	Functions
	Organizar un proceso completo	Logic Apps

# Crear aplicaciones más rápidas

Una vez que la aplicación esté funcionando en Azure, quieres que su rendimiento sea lo más alto posible. Azure proporciona una gama de servicios que pueden ayudarte con eso.

## Azure Content Delivery Network

Uno de los servicios de Azure que pueden ayudarte a hacer tu aplicación más rápida es [Azure Content Delivery Network](#). Sube tus archivos estáticos (vídeos, imágenes, JavaScript, CSS e incluso archivos estáticos HTML) a un almacén de datos como Azure Blob Storage y luego le asocia Azure Content Delivery Network. Content Delivery Network replicará a continuación esos archivos estáticos en cientos de puntos de presencia (PoP) repartidos por todo el mundo. Solo tienes que cambiar la referencia a los archivos estáticos en tu aplicación a una dirección URL diferente. Por ejemplo, la referencia que antes era ~/images/image.png, ahora será https://example.azureedge.com/image.png. Es algo muy fácil de hacer y mejora el rendimiento de la aplicación de las siguientes formas:

- Reduce el contenido suministrado desde la aplicación. Ahora lo suministra Content Delivery Network, lo que libera ciclos de procesamiento en la aplicación
- Lleva el contenido estático físicamente más cerca de tus usuarios distribuyéndolo a puntos de presencia repartidos por todo el mundo

Content Delivery Network puede resultarte útil en aplicaciones web, pero también en aplicaciones móviles y de escritorio.

Un ejemplo de uso de Content Delivery Network es servir vídeos para una aplicación móvil. Los vídeos pueden ser grandes y no deseas almacenarlos en el dispositivo móvil (tampoco lo desean los usuarios). Con Content Delivery Network, se sirven desde el punto de presencia, lo que también mejora el rendimiento, ya que está cerca del usuario.



**Pruébalo** [Empieza a usar Azure Content Delivery Network.](#)

## Azure Redis Cache

Todas las aplicaciones modernas trabajan con datos. Recuperar datos desde un almacén de datos, como una base de datos, conlleva normalmente explorar múltiples tablas o documentos que se encuentran en algún servidor distante, combinar los resultados y enviar el resultado al dispositivo que inició la solicitud. Esto, por supuesto, lleva tiempo y puede frustrar y molestar a tus usuarios.

Para eliminar algunos de estos “recorridos”, puedes almacenar en caché los datos que no cambian a menudo. De esta forma, en lugar de consultar la base de datos cada vez, podrías recuperar algunos de los datos desde una caché, como [Azure Redis Cache](#). La ventaja de la caché es que almacena los datos en un formato sencillo como clave-valor. No necesitas ejecutar una consulta compleja para obtener estos datos, basta con conocer la clave para recuperar el valor. Esto puede mejorar mucho el rendimiento de la aplicación. Así funciona este flujo de trabajo:

1. La aplicación necesita algunos datos e intenta recuperarlos de la caché.
2. Si los datos no están, los obtiene de la base de datos y los almacena también en la caché.
3. La próxima vez que la aplicación busque ese dato, lo encontrará en la memoria caché, lo que evitará un viaje a la base de datos.



Azure proporciona caché como servicio con Redis Cache. Esto se basa en el proyecto de código abierto Redis y cuenta ahora con el respaldo de los ingenieros y los SLA de Microsoft. Es muy eficaz e incluye opciones avanzadas como la agrupación en clústeres y la replicación geográfica.



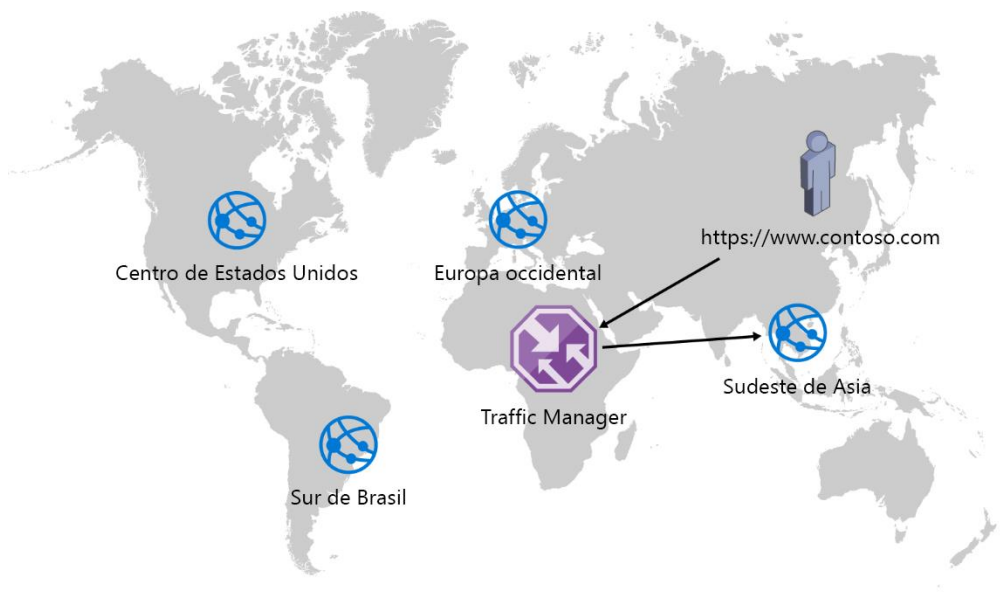
**Pruébalo** [Empieza a trabajar con Azure Redis Cache.](#)

## Azure Traffic Manager

Muchas aplicaciones modernas tienen usuarios en todo el mundo. Proporcionar una experiencia de alto rendimiento para todo el mundo es, como poco, un reto. El problema más obvio al que tendrá que enfrentarse es la latencia. La latencia es el tiempo que tarda una señal o una solicitud en viajar hasta un usuario. Cuanto más lejos esté el usuario de tu aplicación, mayor será la latencia que experimente.

[Azure Traffic Manager](#) ajusta la escala entre distintas regiones, lo que ayuda a reducir la latencia y proporciona a los usuarios una experiencia superior, estén donde estén. Traffic Manager es un mecanismo de enrutamiento inteligente que se antepone a, por ejemplo, las aplicaciones de Aplicación web en todo el mundo. Aplicación web actúa como punto de conexión, y Azure Traffic Manager supervisa el estado y el rendimiento. Como se muestra en la figura 2-2, cuando un usuario accede a tu aplicación, Traffic Manager le dirige a la aplicación de Aplicación web que es más eficaz en su proximidad.

Incluir Traffic Manager en su arquitectura es una forma excelente de mejorar el rendimiento de tu aplicación.



**Figura 2-2:** Azure Traffic Manager dirige el tráfico al punto de conexión con mayor rendimiento desde el punto de vista geográfico

# Dónde almacenar los datos

Los datos son un aspecto muy importante de cualquier aplicación moderna y tienen todo tipo de formas y tamaños. Azure proporciona muchos tipos de almacenes de datos que pueden ayudarte a mantener y recuperar los datos en cualquier escenario. La tabla 2-2 presenta las opciones de almacenamiento disponibles en Azure.

Tabla 2-2: opciones de almacenamiento de Azure

	Bases de datos SQL	MySQL	PostgreSQL	Cosmos DB	Blob	Tabla	Cola	Archivo	Disco	Data Lake Store	SQL Data Warehouse
Datos relacionales	X	X									
Datos relacionales de objetos			X								
Datos no estructurados				X	X						
Datos semiestructurados						X					
Mensajes en cola							X				
Archivos en disco								X			
Archivos de alto rendimiento en disco									X		
Almacenamiento de datos de gran tamaño					X			X	X	X	X
Almacenamiento de datos pequeños	X	X	X	X		X	X	X	X		
Replicación de datos geográficos	X			X							



**Nota** Puedes utilizar casi todas las opciones de almacenamiento mencionadas en esta sección [como activadores y enlaces para Azure Functions](#).

Veamos con más detenimiento cada opción de almacenamiento.

## Azure SQL Database

Si deseas utilizar tablas con filas y columnas para almacenar datos, [Azure SQL Database](#) es una gran opción. Es un sistema de base de datos relacional que es similar a Microsoft SQL Server on-premises. Como SQL Database se ejecuta en el cloud, está totalmente administrado, ofrece un alto rendimiento y copias de seguridad automáticas y es escalable, además de contar con muchas funciones avanzadas.

Puedes utilizar SQL Database con tus herramientas favoritas, como SQL Server Management Studio y Entity Framework.

A continuación se describen algunas de sus características más avanzadas:

- Replicación geográfica, que replica los datos en otras regiones geográficas en tiempo real ([Introducción a la replicación geográfica](#))

- Enmascaramiento dinámico de datos, que enmascara dinámicamente los datos confidenciales para determinados usuarios en tiempo de ejecución ([Introducción al enmascaramiento dinámico de datos](#))
- Auditoría, que proporciona una auditoría completa de todas las acciones que se realizan en los datos ([Introducción a la auditoría de Azure SQL Database](#))

Las bases de datos de SQL Database son extremadamente fiables y sólidas y ofrecen un acuerdo de nivel de servicio (SLA) que garantiza un tiempo de actividad del 99,99 %.

## Azure Database for MySQL

MySQL es un sistema de base de datos relacional de código abierto que utilizan millones de aplicaciones como WordPress, Joomla y Drupal. Ahora, MySQL está disponible como servicio de base de datos administrado en el servicio [Azure Database for MySQL](#). Esto significa que puedes empezar a trabajar rápidamente sin necesidad de configurar o mantener servidores. Azure lo hace todo por ti. Obtienes seguridad y escalabilidad “out of the box”. Además, el servicio Database for MySQL ofrece la misma fiabilidad que SQL Database y proporciona un tiempo de actividad del 99,99 %, líder en el sector.

Como en el caso de SQL Database y Azure Database for PostgreSQL, se realizan copias de seguridad incrementales cada cinco minutos y copias de seguridad completas cada hora, lo que permite recuperar los datos en el estado que tenían hasta 35 días antes.



**Pruébalo** [Crea una base de datos de MySQL y conéctate a ella en Azure.](#)

## Azure Database for PostgreSQL

[Azure Database for PostgreSQL](#), basado en la popular tecnología de base de datos PostgreSQL, proporciona un servicio de bases de datos relacionales de objeto. Esto es ligeramente diferente a los servicios de base de datos relacional, ya que las bases de datos relacionales de objeto pueden almacenar tipos de datos más complejos, como los tipos de datos anidados, y son por tanto más adecuadas para los lenguajes de programación orientada a objetos que utilizamos en nuestras aplicaciones.

Database for PostgreSQL es un servicio administrado que tiene las mismas características que Database for MySQL. Se puede aumentar o reducir su escala, ofrece un alto nivel de disponibilidad (SLA del 99,99 %) y crea copias de seguridad automáticas.



**Pruébalo** [Crea una instancia de Azure Database for PostgreSQL usando la CLI de Azure.](#)

## Azure Cosmos DB

[Azure Cosmos DB](#) es la nueva versión y la nueva marca de DocumentDB. Y mucho más. Cosmos DB es un nuevo tipo de base de datos pensada realmente para el cloud. Estas son algunas de sus principales características:

- SLA del 99,99 % que incluye bajas latencias (menos de 10 ms en las lecturas; menos de 15 ms en las escrituras).
- Replicación geográfica, que replica los datos en otras regiones geográficas en tiempo real ([Cómo distribuir los datos globalmente con Azure Cosmos DB](#)).
- Traffic Management, que envía a los usuarios a la réplica de datos más próxima a ellos.

- Escala global sin límites; solo pagas por el rendimiento y el almacenamiento que necesitas.
- Indexación automática de los datos. Ya no es necesario mantener ni ajustar la base de datos.

Además de todas estas características, Cosmos DB ofrece diferentes API con las que puedes almacenar y recuperar datos, como SQL, JavaScript, Gremlin, MongoDB y Almacenamiento de tablas de Azure. Las distintas API gestionan los datos de diferentes formas. Puedes utilizar documentos como datos y también tablas no estructuradas, gráficos, blobs, etc. Tú utilizas la API que se adapte a tus necesidades y Cosmos DB se encarga del resto. Disfrutarás de rendimiento de nivel de cloud, escalabilidad y fiabilidad, y podrás seguir usando el modelo de programación al que estás acostumbrado.



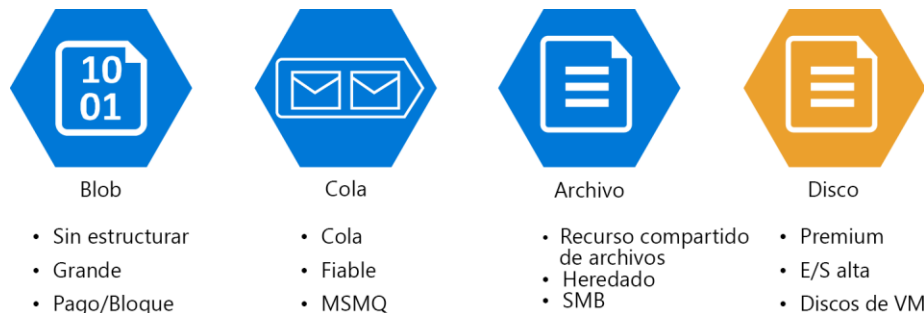
**Pruébalo** [Empieza a usar la API DocumentDB de Azure Cosmos DB.](#)

## Azure Storage

Otra opción para almacenar datos es [Azure Storage](#). Este es uno de los servicios más antiguos, eficaces y fiables de Azure. Azure Storage ofrece cinco tipos de almacenamiento que tienen las siguientes características comunes:

- Redundancia geográfica que replica los datos en distintos centros de datos para que puedas recuperarlos si un centro de datos deja de funcionar en caso de desastre
- Cifrado de datos en tiempo de ejecución
- Dominios personalizados

Los cuatro tipos de Azure Storage son *Blob*, *cola*, *archivo* y *disco* (figura 2-3).



**Figura 2-3:** información general de los servicios de Azure Storage

## Blob Storage

Puedes usar [Azure Blob Storage](#) para almacenar datos no estructurados de gran tamaño, lo que se conoce como "blobs". Pueden ser archivos de vídeo, imagen, audio o texto o incluso archivos de disco duro virtuales (VHD) para máquinas virtuales.

Hay dos tipos de blobs: [blobs de página y de bloque](#). Los blobs de página están optimizados para las operaciones aleatorias de lectura y escritura. Son perfectos para el almacenamiento de un disco duro virtual. Los blobs de bloques están optimizados para cargar eficientemente grandes cantidades de datos. Son idóneos para almacenar archivos de vídeo grandes que no cambian con frecuencia.



**Pruébalo** [Empieza a usar Azure Blob Storage.](#)

## Queue Storage

[Azure Queue Storage](#) es un tipo poco usual de almacenamiento que se utiliza para almacenar pequeños mensajes de datos, pero su principal objetivo es servir como cola. Los mensajes se ponen en la cola y otros procesos los toman de ella. [Este patrón](#) desvincula el remitente del procesador de mensajes y mejora el rendimiento y la fiabilidad. Azure Queue Storage se basa en Microsoft Message Queueing, que se puede encontrar en versiones anteriores de Windows.



**Pruébalo** [Empieza a usar Azure Queue Storage.](#)

## File Storage

Puedes usar [Azure File Storage](#) como una unidad desde la que se comparten los archivos. Usa el protocolo de bloque de mensaje de servidor (SMB), lo que significa que se puede utilizar con Windows y Linux, y se puede tener acceso a él desde el cloud o desde sistemas on-premises. Igual que los otros tipos de Azure Storage, File Storage es escalable y económico.



**Pruébalo** [Empieza a usar Azure File Storage.](#)

## Disk Storage

[Azure Disk Storage](#) es similar a File Storage, pero está pensado específicamente para un alto rendimiento de E/S. Es perfecto para usarlo como unidad en una máquina virtual que necesite alto rendimiento, por ejemplo una máquina virtual en la que se ejecute SQL Server. Sin embargo, hay que tener en cuenta que Disk Storage solo está disponible en el nivel de precios premium de Azure Storage.

## Azure Data Lake Store

Los almacenes de datos anteriores estaban diseñados para el uso normal con aplicaciones o para el uso con máquinas virtuales. [Azure Data Lake Store](#) es un almacenamiento para aplicaciones de Big Data. Se usa para almacenar grandes cantidades de datos en su formato nativo: estructurados, no estructurados y cualquier tipo intermedio. La finalidad de Data Lake Store es albergar los datos sin procesar para poder analizarlos o transformarlos y moverlos. A continuación se muestran las principales características de Azure Data Lake:

- Capacidad de almacenamiento ilimitada. Un archivo puede tener más de un petabyte: 200 veces más de lo que ofrecen otros proveedores del cloud.
- Rendimiento escalable para permitir grandes cantidades de análisis en paralelo.
- Puedes almacenar datos en cualquier formato sin un esquema.

Este es un enfoque muy diferente al del almacén de datos tradicional, en el que los esquemas de datos se definen de antemano.

Por ejemplo, podrías utilizar Data Lake Store para almacenar todos los datos que obtienes de los dispositivos del Internet de las cosas (IoT) que están recopilando datos de temperatura. Puedes dejar los datos en el almacén y luego filtrarlos y crear una vista de los datos por hora o por semana. Almacenar los datos en Data Lake Store es bastante económico, así que se pueden guardar años de datos en él a un coste muy bajo.



**Pruébalo** [Introducción a Azure Data Lake Store con Azure Portal.](#)

## Azure SQL Data Warehouse

Puedes usar [Azure SQL Data Warehouse](#) si necesitas una solución de almacenamiento de datos tradicional totalmente administrada, escalable en tamaño, segura y de alto rendimiento. Los datos se almacenan en esquemas predefinidos y se consultan usando el lenguaje SQL de SQL Server ya conocido.

Dado que SQL Data Warehouse se ejecuta en Azure, permite el uso de características avanzadas como la detección automática de amenazas, que utiliza Machine Learning para entender los patrones de la carga de trabajo y servir como sistema de alarma para avisar de posibles incumplimientos.

Un ejemplo de uso de SQL Data Warehouse es cuando se sabe qué informes se van a mostrar a los usuarios y cuál es el esquema de datos de esos informes. En ese caso, puedes crear esquemas en SQL Data Warehouse y rellenarlo con los datos para que los usuarios pueden examinar los datos.



**Pruébalo** [Crea un Azure SQL Data Warehouse.](#)

# Agregar inteligencia a tu aplicación

Hospedar tu aplicación y tus datos y lograr que sea escalable y segura y que tenga un alto rendimiento está muy bien, pero ¿no sería estupendo poder agregar características inteligentes a la aplicación?

## Búsqueda de Azure

La búsqueda es una característica habitual en la mayoría de las aplicaciones, y sin embargo siempre ha sido una función difícil de implementar. [Búsqueda de Azure](#) proporciona una buena parte de la base necesaria para hacer búsquedas. Solo tienes que usar una instancia de Búsqueda de Azure, crear un índice que te ayude a buscar y rellenarlo con datos, eso es todo. Hay muchas opciones que puedes usar para adaptar Búsqueda de Azure y hay un montón de estupendas características que facilitarán las búsquedas a los usuarios:

- Búsqueda geográfica que permite a los usuarios explorar datos en función de la proximidad de un resultado de búsqueda a una ubicación física.
- Los analizadores de lenguaje de Lucene, así como los procesadores de lenguaje natural de Microsoft (NLP), disponibles en 56 idiomas para gestionar de forma inteligente la lingüística de cada idioma, incluidos los tiempos verbales, los sustantivos con plurales irregulares (por ejemplo, “vez” frente a “veces”), la descomposición de palabras, la división de palabras (para los idiomas sin espacios) y mucho más.
- Supervisión y creación de informes que indican lo que se ha buscado, lo rápida que ha sido la búsqueda y si ha tenido éxito.

- Características de experiencia del usuario, como ordenar y paginar los resultados de búsqueda, filtrado inteligente y sugerencias de búsqueda.

Podrías implementar Búsqueda de Azure para ayudar a los usuarios a realizar búsquedas en tu catálogo de productos en una aplicación de comercio electrónico.



**Pruébalo** [Crea tu primer índice de Búsqueda de Azure en el portal.](#)

## Azure Cognitive Services

Con [Azure Cognitive Services](#), puedes agregar increíbles funciones inteligentes a tus aplicaciones. Cognitive Services es una colección de API que puedes utilizar para lograr cosas que a veces parecen pura magia. Hay más de 25 API en las categorías de visión, habla, lenguaje, conocimiento y búsqueda. Aquí puedes ver las API disponibles:

- Con [Face API](#), puedes realizar reconocimiento e identificación facial e incluso determinar el color de ojos de una persona, aunque esa persona lleve gafas.
- [Emotion API](#) determina las emociones que una persona muestra en una foto.
- [Language Understanding Intelligent Services API](#) (o LUIS) puede entender texto a partir del modelo que se use para entrenarla.
- [Translator Speech API](#) convierte la voz en texto y lo traduce en tiempo real

Cognitive Services proporciona estas funciones inteligentes a través de Machine Learning e inteligencia artificial, que puedes entrenar con tus propios conjuntos de datos. Estos servicios son sencillamente increíbles y abren posibilidades con las que no podíamos siquiera soñar hace tan solo unos años.

## Azure Bot Service

[Azure Bot Service](#) hace que sea fácil crear un *bot*, un software que puede interactuar con los usuarios de forma automática y autónoma.

Crear un bot no es una tarea trivial. Necesitas hacer un seguimiento del contexto de su interacción con el usuario y debes estar preparado para responder a muchos parámetros de interacción posibles. Bot Service te ayuda a hacerlo de las siguientes formas:

- Te ayuda a realizar un seguimiento del contexto de interacción y te proporciona plantillas para comenzar desde Bot Framework.
- Se integra perfectamente con Cognitive Services, lo que te permitirá crear un bot "inteligente".
- Esto te ayudará a integrarlo con servicios como Facebook, Slack, Microsoft Teams, Telegram, etc.
- Ofrece toda las ventajas de un servicio administrado en Azure: gran escalabilidad, entrega continua (CD) integrada y pagar solo por lo que se usa.

Por ejemplo, con Bot Service puedes crear un bot que proporcione a los usuarios respuestas a sus preguntas más frecuentes. La interfaz del bot puede ser un cuadro de chat que esté en tu sitio web.



**Pruébalo** [Empieza a trabajar con un bot de chat usando Azure Bot Service.](#)



# Usar eventos y mensajes en una aplicación

Las aplicaciones modernas distribuidas globalmente deben lidiar a menudo con grandes cantidades de mensajes y se deben diseñar pensando en la desvinculación y el escalado. Azure proporciona varios servicios para ayudar con la ingesta y el análisis de eventos y los patrones de mensajería.

## Azure Service Bus

En Azure, la base de la mensajería es [Azure Service Bus](#). El servicio Service Bus incluye una colección de servicios que se pueden utilizar para los patrones de mensajería. Entre ellos destacan Colas y Temas de Azure Services Bus.



**Pruébalo** Introducción a [Colas](#) y [Temas](#) de Azure Service Bus.

### Colas de Azure Service Bus

Las [colas de Azure Service Bus](#) se usan para desacoplar sistemas. Por ejemplo: una aplicación web recibe pedidos de los usuarios y tiene que invocar un servicio web para procesar el pedido. El servicio web tardará mucho tiempo, quizás hasta cinco minutos, en procesar completamente el pedido. Por supuesto, es inaceptable para la aplicación web esperar cinco minutos entre el momento en el que el usuario realiza el pedido y el momento en el que se muestran los comentarios. Una buena forma de solucionar este problema es utilizar una cola para desacoplar la aplicación web del servicio web.

La aplicación web recibe el pedido y lo escribe en un mensaje en una cola de Azure Service Bus. Una vez que lo ha hecho, puede informar al usuario de que se está tramitando el pedido. El servicio web toma los mensajes de la cola, uno por uno, y los procesa. Cuando el servicio web termina de procesar un pedido, envía una notificación por correo electrónico al usuario para informarle de que se ha pedido el artículo.

Al desacoplar los sistemas, la aplicación web puede funcionar a una velocidad diferente del servicio web, y ambos pueden escalarse por separado para ajustarse a las necesidades de las aplicaciones.

La cola de Service Bus es un mecanismo sencillo. Varias aplicaciones pueden poner mensajes en la cola, pero un mensaje de la cola solo puede procesarse en una aplicación cada vez. Incluye algunas características inteligentes para trabajar con los mensajes de la cola, como la detección de duplicados y una subcola de mensajes fallidos a la que se transfieren los mensajes cuando no se pueden procesar correctamente.

### Temas de Azure Service Bus

Al igual que las colas de Service Bus, los [temas de Azure Service Bus](#) proporcionan una forma de desacoplar aplicaciones. Estas son las diferencias entre la cola y los temas:

- Con una cola, varias aplicaciones escriben mensajes en la cola, pero *solo una aplicación* puede procesar un mensaje en un momento dado.
- Con un tema, varias aplicaciones escriben mensajes en el tema y *varias aplicaciones* puede procesar un mensaje al mismo tiempo.

Los temas de Service Bus funcionan como una cola y varias aplicaciones pueden procesar el mismo mensaje. Las aplicaciones pueden crear una suscripción en el tema que indica qué tipo de mensajes les interesa. Al igual que las colas, los temas tienen características como la detección de duplicados y una subcola de mensajes fallidos a la que los mensajes se mueven cuando no pueden procesarse correctamente.

### Comparación entre las colas de Service Bus y las colas de Azure Storage

Hay colas de Service Bus, pero también hay colas de Azure Storage. Básicamente hacen lo mismo, pero hay diferencias sutiles entre ellas. La tabla 3-1 analiza los dos tipos de colas.

**Tabla 3-1:** comparación de características entre las colas de Service Bus y las colas de Azure Storage

Colas de Azure Service Bus	Colas de Azure Storage
Duración del mensaje > 7 días	Duración del mensaje < 7 días
Orden garantizado (el primero en entrar es el primero en salir)	Tamaño de la cola > 80 GB
Detección de duplicados	Registros de transacciones
Tamaño del mensaje ≤ 1 MB	Tamaño del mensaje ≤ 64 KB

## Azure Event Hubs

En un mundo como el nuestro, donde el software es algo ubicuo, se generan datos desde innumerables aplicaciones, dispositivos y ubicaciones todo el tiempo. Hoy en día, es cada vez más común para las compañías capturar enormes cantidades de esos datos para analizarlos o transformarlos y moverlos para su uso posterior. [Azure Event Hubs](#) puede ayudar con esto.

Event Hubs se ha diseñado para la ingesta masiva de datos. Puedes lanzar millones de mensajes por segundo al servicio y este gestionará los datos sin esfuerzo. Puedes retener mensajes hasta un máximo de siete días o retenerlos indefinidamente escribiéndolos en un almacén de datos mediante la función de captura de Event Hubs.

Puedes utilizar Event Hubs para filtrar los datos con consultas en el momento en que llegan y enviarlos a un almacén de datos como Azure Cosmos DB. Puedes incluso reproducir mensajes si es necesario.



**Pruébalo** [Empieza a enviar mensajes a Azure Event Hubs en .NET Standard.](#)

## Azure IoT Hub

Al igual que Event Hubs, [Azure IoT Hub](#) está diseñado para la ingesta masiva de datos. Va dirigido en especial a la gestión del enorme volumen de mensajes de datos procedentes de los dispositivos del Internet de las cosas (IoT; consulta la barra lateral del capítulo 1, "¿Quién usa Azure?", para ver un ejemplo), como los termostatos inteligentes o los sensores de los coches. Tiene muchas de las mismas propiedades que Event Hubs, como la capacidad de retener mensajes hasta un máximo de siete días y de reproducirlos.

Lo que hace único a IoT Hub es que, además de recibir mensajes de los dispositivos, puede enviarles mensajes. Tiene la capacidad de administrar toda la infraestructura de IoT. Puedes usar IoT Hub para registrar dispositivos e informar de su estado y para administrarlos protegiéndolos y reiniciándolos y enviándoles datos.



**Pruébalo** [Conecta tu dispositivo a tu IoT Hub con .NET.](#)

## Azure Event Grid

[Azure Event Grid](#) ofrece un tipo diferente de mensajería. Event Grid ofrece un servicio de publicación/suscripción totalmente administrado, que conecta con casi todos los servicios de Azure y puede conectar con suscriptores y publicadores personalizados.

Esto es diferente de trabajar con las colas y los temas de Service Bus, donde solo se necesita sondear la cola o el tema para extraer nuevos mensajes. Event Grid envía automáticamente los mensajes a los suscriptores, lo que lo convierte en un servicio de eventos reactivo en tiempo real.

Da servicio dentro y fuera de los eventos de publicación de Azure (por ejemplo, cuando se agrega un nuevo Blob o cuando se agrega un nuevo usuario a una suscripción de Azure). Azure Event Grid puede detectar estos eventos y hacer que estén disponibles para los controladores de eventos y servicios que se han suscrito a los eventos (Figura 3-1). Los controladores de eventos pueden ser cosas como Azure Functions o Azure Logic Apps, que pueden actuar sobre los datos del evento.

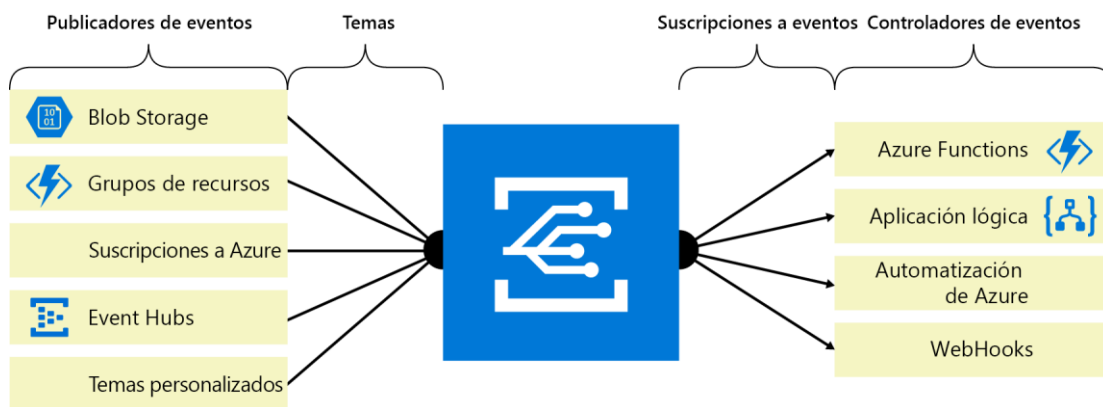


Figura 3-1: información general de Azure Event Grid

Otro aspecto importante de Event Grid es que es un servicio sin servidor. Esto significa que, al igual que Azure Logic Apps y Azure Functions, Event Grid ajusta su escala automáticamente y no es necesario implementar una instancia. Solo tienes que configurarlo y usarlo. Y solo se paga por lo que se usa, no por el hecho de que esté ahí.

Veamos un ejemplo de uso de Azure Event Grid:

Deseas recibir una notificación por correo electrónico cada vez que se agregue o se elimine un usuario en tu lista de correo de MailChimp.

Azure Event Grid se utiliza para activar una aplicación en Azure Logic Apps y está configurado para escuchar los cambios que se producen en la lista de correo de MailChimp. El siguiente paso es enviar un correo electrónico que contiene el nombre del usuario que se ha agregado o eliminado y la acción que se realizó (agregar o eliminar). Ahora, cuando un nuevo usuario se agrega a la lista de correo, Event Grid transmite una señal a Logic Apps, que envía un correo electrónico.



**Pruébalo** [Supervisa los cambios de las máquinas virtuales con Azure Event Grid y Logic Apps.](#)

## ¿Qué se debe usar en cada caso?

Azure proporciona numerosas opciones de mensajería y para desacoplar aplicaciones. ¿Cuál se debe utilizar y cuándo? La tabla 3-2 resume las diferencias para ayudarle a clasificarlas.

**Tabla 3-2:** cuándo utilizar cada servicio de Azure para mensajes o eventos

	Event Grid	Event Hubs	IoT Hub	Temas	Colas de Service Bus	Colas de Storage
Ingesta de eventos	X	X	X			
Administración de dispositivos			X			
Mensajería	X	X	X	X	X	X
Varios consumidores	X	X	X	X		
Varios remitentes	X	X	X	X	X	X
Uso para desacoplar		X	X	X	X	X
Uso para publicar/suscribir	X					
Tamaño máximo del mensaje	64 KB	256 KB	256 KB	1 MB	1 MB	64 KB

# Proteger una aplicación

Hoy en día la seguridad es uno de los aspectos más importantes de cualquier aplicación y no es fácil gestionarla bien. Afortunadamente, Azure ofrece muchos servicios que te ayudarán a proteger tu aplicación. En este capítulo veremos algunos de esos servicios.

## Azure Active Directory

Una parte importante de la seguridad de tu aplicación es la autenticación de los usuarios antes de utilizarla. La autenticación no es fácil de implementar. Hay que almacenar las identidades y las credenciales de los usuarios en algún sitio, implementar la administración de contraseñas, crear un protocolo de enlace de autenticación seguro, etc.

[Azure Active Directory](#) proporciona todo esto y más "out of the box". Las identidades de los usuarios se guardan en Azure Active Directory y los usuarios se autentican en él, de modo que solo se les redirige a la aplicación una vez que se han autenticado. Azure Active Directory se encarga de administrar las contraseñas, lo que incluye situaciones frecuentes como "He olvidado la contraseña".

Millones de aplicaciones usan Azure Active Directory cada día, entre ellas [Azure Portal](#), [Outlook.com](#) y [Office 365](#). Debido a esto, puede detectar con más facilidad el comportamiento malicioso y responder a él. Por ejemplo, si un usuario inicia sesión en una aplicación desde una ubicación de Europa y un minuto más tarde inicia sesión desde Australia, Azure Active Directory marca esto como un comportamiento malintencionado y pide al usuario credenciales adicionales por medio de una autenticación multifactor.

## Azure API Management

Las API debe ser seguras. Esto es cierto tanto para las API que creas tú mismo como para las que consumes de otros proveedores. Para ayudar a proteger tus API, puedes utilizar [Azure API Management](#). Es básicamente un proxy que antepone a tus API o a las de otro proveedor y que agrega características como el almacenamiento en caché, la limitación y la autenticación y autorización.

Puedes proteger tu API con API Management pidiendo a los consumidores que se suscriban a ella. De esta forma, las aplicaciones que llaman necesitan autenticarse para poder usar tu API. Puedes utilizar varios métodos de autenticación, como tokens de acceso, autenticación básica y certificados. Además, puedes realizar un seguimiento para ver quién llama a tu API y bloquear a los autores de llamadas no deseados.



**Pruébalo** [Introducción a Azure API Management.](#)

## Azure Key Vault

Dentro de tu arquitectura de seguridad, necesitas un lugar seguro para almacenar certificados, claves y otros secretos. [Azure Key Vault](#) lo proporciona. Con Key Vault, tienes una ubicación central en la que se almacenan los secretos utilizados por tus aplicaciones.

Pueden ser las credenciales de una `connectionstring` utilizada por tu aplicación. Tu aplicación obtendría la `connectionstring` de Key Vault en vez de obtenerla desde el sistema de configuración. De esta forma, los administradores pueden controlar los secretos y los desarrolladores no tienen que ocuparse de ellos.

Key Vault también se utiliza para almacenar los certificados, como Capa de sockets seguros (SSL), que se usan para proteger el tráfico intercambiado por las aplicaciones a través de HTTPS.



**Pruébalo** [Introducción a Azure Key Vault.](#)

## Protección de Azure contra ataques de denegación de servicio distribuidos

Lo has oído demasiadas veces en las noticias y sin duda no deseas que le ocurra a tu compañía: tu aplicación recibe un ataque de denegación de servicio distribuido (DDoS). Estos tipos de ataques son cada vez más frecuentes y pueden desbordar a una aplicación hasta el punto de que nadie pueda volver a usarla.

Azure te protege frente a ataques de DDoS con su capa de protección de DDoS de Azure. Esta es una capa de la red física de Azure que no es accesible a los clientes. Forma parte de la plataforma Azure y se obtiene sin coste con todos los servicios. No tienes que hacer nada ni configurar ningún ajuste para usarla. Siempre está ahí para protegerte.



**Más información** [Más información sobre la protección de Azure frente a DDoS.](#)

## Firewall de aplicaciones web de Azure

Necesitas proteger tu aplicación frente a muchas posibles amenazas, incluidas las definidas en la especificación de Open Web Application Security Project (OWASP), como la inyección SQL y Cross-Site Scripting (XSS). El [firewall de aplicaciones web de Azure](#) puede ayudarte con eso. El firewall de aplicaciones web, una característica del servicio [Azure Application Gateway](#), proporciona protección en tiempo real para tu aplicación.

Puede detectar un ataque malicioso, tal y como se define en el [conjunto de reglas básicas de OWASP](#), e impedir que el ataque llegue a tu aplicación. También informa sobre los ataques que se han intentado o están en curso para que puedas ver qué amenazas están tratando de llegar a tu aplicación. Esto proporciona una capa adicional de seguridad que protege tus aplicaciones.

## Cifrado predeterminado de datos

Para proteger tus almacenes de datos lo mejor posible, defines `connectionstrings` y reglas de firewall. Pero si un atacante consigue obtener acceso a tus datos de algún modo, puede lograr que el ataque sea inútil por medio del cifrado.

De manera predeterminada, los datos se cifran cuando se almacenan en [Azure SQL Database](#), [Azure SQL Data Warehouse](#), [Azure Database for MySQL](#), [Azure Database for PostgreSQL](#), [Azure Storage](#), [Azure Cosmos DB](#) o [Azure Data Lake Store](#). Y todo esto funciona automáticamente; no tienes que hacer nada para configurarlo ni para cifrar o descifrar los datos al utilizarlo.

# Dónde y cómo implementar los servicios de Azure

Los servicios de Azure con los que elijas trabajar determinan tus opciones de implementación y viceversa. Por tanto, es importante entender las opciones de implementación que hay en Azure y saber cuáles son sus ramificaciones.

## El punto de vista de DevOps

Entregar rápido y reiterar rápido son aspectos cruciales para crear un gran software. Por tanto, el nuevo código debe fusionarse con el código de tu equipo, y eso hay que implementarlo y probarlo tan deprisa y tan a menudo como sea posible para ver si todo funciona y si lo que se ha creado es lo que necesita el usuario.

Muchos de los servicios de Azure que hemos analizado hasta ahora en este libro admiten la entrega automática de código, a menudo a través de un proceso de [entrega continua](#) (CD) que se configura dentro del servicio.



Además de las funcionalidades nativas de los servicios de Azure, puedes utilizar Visual Studio Team Services para crear, probar e implementar la aplicación. [Puedes crear fácilmente nuevos procesos de compilación e implementación en Visual Studio Team Services](#) y hacer cosas como automatizar las pruebas de carga y cambiar las ranuras de implementación a la producción.



**Pruébalo** [Introducción a Visual Studio Team Services.](#)

## Plantillas de Azure Resource Manager

Todos los servicios de Azure descritos en esta guía se basan en Azure Resource Manager. Podemos aprovechar este hecho para convertir nuestro entorno en infraestructura como código (IaC) utilizando [plantillas de Resource Manager](#). Estas plantillas son archivos JSON que describen lo que se desea implementar y especifican los parámetros correspondientes.

Puedes crear plantillas de Resource Manager en Visual Studio y Visual Studio Code usando la plantilla de proyecto de Grupo de recursos de Azure. También puedes generar plantillas de Resource Manager desde Azure Portal haciendo clic en el botón Script de automatización, que está disponible en la barra de menús de todos los recursos disponibles en Azure Portal. Esto genera la plantilla de Resource Manager para el recurso dado e incluso código para crear el recurso utilizando la interfaz de la línea de comandos (CLI) de Azure, Windows PowerShell, .NET y Ruby.

Una vez que tengas una plantilla de Resource Manager, puedes implementarla en Azure usando Windows PowerShell, la CLI de Azure o Visual Studio. También puedes automatizar tu implementación en un proceso de CD usando Visual Studio Team Services.

Utiliza plantillas de Resource Manager para implementar aplicaciones de modo que se ejecuten en la plataforma Azure, ya sea en el cloud público u on-premises, en Azure Stack.

Un gran ejemplo de implementación de recursos en el cloud con Resource Manager es usar el botón Implementar en Azure que se puede encontrar en muchos repositorios de GitHub, como se muestra en la figura 5-1.



**Figura 5-1:** implementación con un botón de Azure Logic Apps usando una plantilla de Resource Manager en GitHub

## Azure Service Fabric

Otra forma de ejecutar tus aplicaciones es ejecutarlas en [Azure Service Fabric](#). Service Fabric es la base que Microsoft utiliza para ejecutar muchos recursos de Azure y lograr que tengan un alto grado de disponibilidad y de rendimiento y que se reparen de forma automática. Ahora puedes usar Service Fabric tú mismo para hospedar tus propios servicios.

Puedes utilizar el SDK de Azure Service Fabric para crear aplicaciones para Service Fabric. También puedes ejecutar cualquier archivo ejecutable en Service Fabric e incluso usarlo para hospedar contenedores.

Service Fabric tiene la capacidad de hacer que tus aplicaciones disfruten del mismo nivel de rendimiento, fiabilidad y seguridad que muchos servicios de Azure y puedes usarlo en cualquier lugar: puedes implementar Service Fabric en Azure, on-premises, en tu propio equipo e incluso en máquinas virtuales (VM) de otros clouds.

Una vez que implementes tu aplicación en Azure Service Fabric, te proporcionará ventajas como las siguientes:

- Equilibrio de carga
- Escalado automático
- Alta disponibilidad
- Reparación automática
- Replicación y conmutación por error
- Distribución de actualizaciones
- Y reversión automática

## Contenedores de Azure

“Contenedores” es una de esas palabras tecnológicas de moda que aparecen en todas las noticias. Pero no es solo una palabra, son de hecho muy útiles para ejecutar las aplicaciones. Un contenedor es básicamente una máquina virtual ligera que se inicia y se detiene mucho más deprisa que una máquina virtual y por tanto es mucho más útil para desarrollar, probar y ejecutar aplicaciones en producción.

El mayor beneficio que se deriva de los contenedores es que un contenedor es siempre igual. Un contenedor se ejecuta localmente al desarrollar la aplicación y se utiliza la misma configuración del contenedor en el cloud o en cualquier otro lugar. Tu equipo utiliza exactamente la misma configuración de contenedor, así que sabes que la infraestructura es la misma para todo el mundo y en la producción. Con los contenedores, la excusa típica de los desarrolladores, “en mi equipo sí funciona”, ahora significa que también funcionará en la producción.

Hay muchas tecnologías para ejecutar contenedores, y [Docker](#) es una de ellas. Azure puede ejecutar y administrar los contenedores con [Azure Container Instances](#) y [Azure Container Service](#), e incluso en Services Fabric y [Azure App Service en Linux](#).

La tabla 5-1 muestra qué servicio se puede usar en cada caso cuando se trabaja con contenedores.

**Tabla 5-1:** elegir el servicio de Azure que se debe usar para los contenedores

	Azure Container Services	Azure Container Instances	Azure Service Fabric
Para las implementaciones de producción de sistemas complejos (con un orquestador de contenedores)	X		
Para ejecutar configuraciones sencillas (posiblemente sin orquestador)		X	
Para cargas de trabajo con tiempos de ejecución largos en los contenedores	X		
Para cargas de trabajo con tiempos de ejecución cortos en los contenedores		X	

Para organizar un sistema basado en contenedores	X		X
Organización con orquestadores de código abierto ( <a href="#">DC/OS</a> , <a href="#">Docker Swarm</a> , <a href="#">Kubernetes</a> )	X		
Organizar con orquestador integrado			X



**Nota** Recuerda que, al usar contenedores, utilizas un producto de infraestructura como servicio (IaaS) y eres responsable del sistema operativo (SO), las revisiones, el equilibrio de carga, etc.

## Azure Stack

Si necesitas que tus aplicaciones y datos permanezcan on-premises, pero quieres aprovechar la capacidad de Azure, [Azure Stack](#) es el producto que necesitas. Azure Stack, único en el sector, es una extensión de Azure que se hospeda en el entorno del usuario. Esencialmente, es Azure en una caja.

Azure Stack se usa del mismo modo que Azure, con la misma experiencia de Azure Portal y las mismas API, con las que puedes usar la CLI de Azure, Windows PowerShell o tu IDE favorito.

Puedes ejecutar cosas como Azure App Service y Azure Virtual Machines en Azure Stack. Todo es exactamente igual que en el cloud público, pero ahora se ejecuta on-premises. Y si decides cambiar al cloud público, solo tienes que transferir los servicios de Azure Stack a Azure.

## Dónde y cuándo implementar

La tabla 5-2 resume las opciones de implementación para Azure. Ten en cuenta que usar todos los servicios de la plataforma como servicio (PaaS) descritos en esta guía solo es posible cuando se usa el cloud público con plantillas de Resource Manager u on-premises con Azure Stack.

**Tabla 5-2:** comparación de opciones de implementación de Azure

	IaaS	PaaS
On-premises	Azure Stack Contenedores Service Fabric	Azure Stack Service Fabric
Cloud público	Contenedores Service Fabric	Service Fabric Plantillas de Resource Manager

# Tutorial sobre Azure

Ahora que ya sabes lo que es Azure y conoces los servicios que ofrece, vamos a empezar a utilizarlo.

## Tutorial n.º 1: Azure Portal

Una de las herramientas más importantes de Azure es Azure Portal, que se puede encontrar en <https://portal.azure.com/>. Esta es la ubicación central de Azure, en la que puedes hacer todo lo que desees con Azure. La mayoría de las cosas que se pueden hacer en Azure Portal se pueden hacer también a través de la API de Azure, la interfaz de la línea de comandos (CLI) de Azure y Azure PowerShell.

Puedes usar una preview de Azure Portal visitando <https://preview.portal.azure.com>. La preview contiene características que no están disponibles de forma general pero que se pueden probar para proporcionar comentarios sobre ellas. El resto de esta guía se centrará en la versión del portal que está disponible normalmente, no la preview.

Lo primero que verás en Azure Portal es un panel con mosaicos. Puedes crear y personalizar los paneles y compartirlos con los miembros del equipo o reservarlos para tu uso exclusivo.

### Mosaicos en Azure Portal

Los paneles contienen mosaicos que muestran información sobre un servicio o sirven como accesos directos a un servicio. Puedes encontrar estos mosaicos, que se muestran en la figura 6-1, por todo el portal, en las páginas de todos los servicios. Pueden resultar muy útiles para obtener información general sobre el funcionamiento de un servicio con rapidez.

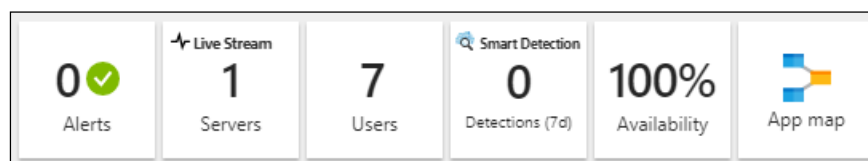


Figura 6-1: mosaicos informativos en Azure Portal

Por ejemplo, el servicio Application Insights muestra de manera predeterminada mosaicos que indican si hay alguna alerta activa, si están llegando datos en directo, cuántos usuarios han estado activos en las últimas 24 horas, cuál ha sido la disponibilidad, etc. Se trata de información muy útil en un formato muy consumible. Puedes personalizar el tamaño y la información de los mosaicos. También puedes personalizar el aspecto de los gráficos ajustando sus escalas de tiempo y haciendo que muestren los datos en diferentes formatos, como líneas o barras.

También puedes anclar mosaicos (figura 6-2) directamente en los paneles para que sean lo primero que se ve al entrar en el portal. Por ejemplo, puedes anclar mosaicos de todas las métricas de servicio que te interesen para crear un panel de supervisión con el fin de [compartirlo con el equipo](#) o de visualizarlo en un monitor en la sala del equipo.

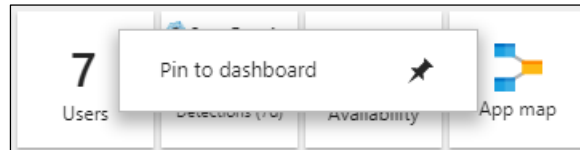


Figura 6-2: puede anclar mosaicos en un panel

## Buscar servicios

Los servicios de Azure son el tema central de Azure Portal. Puedes agregarlos y buscarlos de varias formas.

Para crear nuevos servicios, en la esquina superior izquierda de la ventana del portal, haz clic en el signo más ("+" ). Esto abre el cuadro de búsqueda de Marketplace, donde lo encontrarás todo: desde Aplicación web a Linux Server, como se muestra en la figura 6-3.

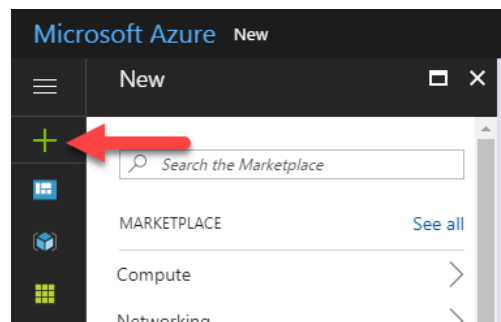


Figura 6-3: botón Nuevo recurso en Azure Portal

Cuando encuentres el servicio que deseas (figura 6-4), un asistente te guiará por su configuración e implementación.

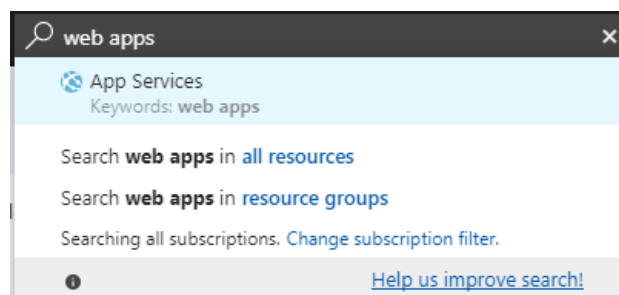


Figura 6-4: resultados de la búsqueda de recursos

Si tienes varios recursos, puedes encontrarlos por medio de la búsqueda. Puedes usar el cuadro de búsqueda de la parte superior del portal (figura 6-5) para buscar en todos tus recursos e ir directamente a ellos.

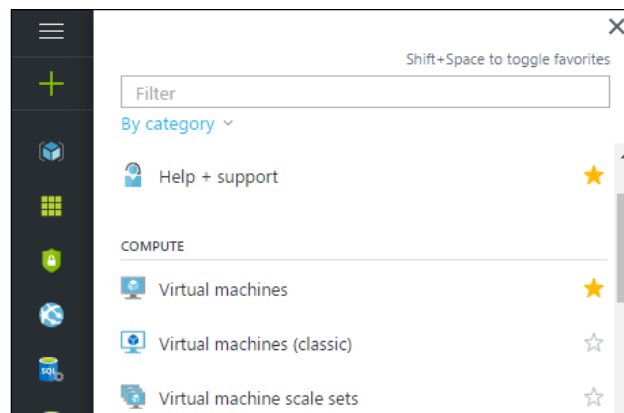


Figura 6-5: menú Favoritos de Azure Portal

En el panel del lado izquierdo del portal se encuentra el menú Favoritos. Este menú muestra las categorías de recursos (representadas por sus iconos) como Azure App Service. Puedes reorganizar los iconos arrastrándolos arriba y abajo. También puedes seleccionar los que desees ver expandiendo el menú Favoritos y haciendo clic en las estrellas de esas categorías.

## Descripción de las hojas

En Azure las páginas se denominan también *hojas*. Las hojas están en todas partes y puedes incluso anclarlas en los paneles. Cuando se abre una aplicación web, lo primero que se ve es la hoja de información general. Esta hoja proporciona herramientas para detener, iniciar y reiniciar la aplicación web y muestra mosaicos con sus métricas, como el número de solicitudes y de errores. Si se elige otro elemento de menú, se abre una nueva hoja. Las hojas siempre se abren en contexto. Así, si abres la hoja Ranuras de implementación y haces clic en el botón Agregar nueva ranura de implementación, aparece una nueva hoja a la derecha de la hoja Ranuras de implementación que indica el contexto en el que te encuentras.

## Crear una nueva máquina virtual

Vamos a usar Azure Portal para crear una nueva máquina virtual (VM) que ejecuta Red Hat Linux. Después de crearla, te mostraremos cómo apagarla y eliminarla para no seguir pagando por ella.

### Una nota sobre los grupos de recursos

La máquina virtual se implementará en un *grupo de recursos*. Todos los recursos de Azure residen en grupos de recursos. Un grupo de recursos es un contenedor lógico que alberga los recursos. Puedes administrar la seguridad de un grupo de recursos y ver lo que cuestan los recursos del grupo. Es una práctica habitual agrupar los servicios relacionados en un grupo de recursos para facilitar su protección y para poder determinar fácilmente lo que cuestan.

1. En Azure Portal, en la esquina superior izquierda, haz clic en el botón Crear un nuevo servicio.
2. En el cuadro de búsqueda, escribe **Linux virtual machine**.

Obtendrás muchos resultados de búsqueda, entre los que se incluye Red Hat Enterprise Linux. Vamos a probar su versión más reciente.

3. Haz clic en la versión más reciente de Red Hat Enterprise Linux.
4. Haz clic en Crear.

Se abre el Asistente para crear máquinas virtuales.

- a. Elige un nombre para la máquina virtual.
- b. Elige el tipo de unidad. SSD proporciona una máquina virtual más rápida, pero también es más caro. Para este tutorial, elige SSD.
- c. Escribe un nombre de usuario.
- d. Selecciona Contraseña en Tipo de autenticación.
- e. Escribe una contraseña y confírmala.
- f. En el cuadro Grupo de recursos, escribe un nuevo nombre.
- g. Elige la ubicación de la máquina virtual y haz clic en Aceptar.

5. Elige el tamaño de la máquina virtual.

El rendimiento de la máquina virtual determina lo que se paga por ella. Hay muchas opciones de tamaño para las máquinas virtuales, algunas son pequeñas y otras increíblemente grandes. Puedes usar el asistente para seleccionar cuántos núcleos y cuánta memoria deseas y elegir las opciones en función de eso. Además, hay otras características que tienen opciones de tamaño:

- a. Tipo de disco duro (SSD o disco duro normal).
- b. Cantidad de IOPS máx. (operaciones de entrada/salida por segundo). Esto determina en gran medida el rendimiento de la máquina virtual, especialmente si las aplicaciones ejecutadas leen y escriben mucho en el disco duro.
- c. Cantidad de unidades de datos que se pueden instalar en la máquina virtual.
- d. Capacidad de equilibrar la carga.
- e. Tarjeta gráfica que se instala en la máquina virtual. Esto resulta útil si se necesitas hacer mucha representación de gráficos o si hay cargas de trabajo computacionales altas.

6. Después de seleccionar el tamaño, puedes configurar ajustes adicionales como la red virtual y la dirección IP e instalar extensiones en la máquina. Por ahora, déjalo todo como está y haz clic en Aceptar.
7. Repasa el resumen, acepta los términos y haz clic en Comprar.

Ahora se implementará la máquina virtual. Generalmente se tarda solo unos minutos.

### Conectar a la máquina virtual a través de Secure Shell

Ahora que has creado una máquina virtual con Red Hat Linux, puedes conectarse a ella y comenzar a utilizarla. Puedes utilizar varias herramientas como Secure Shell (SSH) o incluso un [escritorio remoto en la máquina virtual](#). Vamos a utilizar SSH para conectar a la máquina virtual utilizando Cloud Shell en Azure Portal.

Cloud Shell es la CLI de Azure en el navegador: ya no tienes que instalar nada en tu PC. Cloud Shell utiliza la versión 2.0 de la CLI de Azure. Puedes utilizar Bash o PowerShell en el cloud para interactuar con tus recursos de Azure.

Para iniciar Cloud Shell, en la parte superior derecha de Azure Portal, haz clic en el botón que se muestra a continuación:



Para conectarte a la máquina virtual, necesitas su dirección IP. Busca la máquina virtual en Azure Portal (ya sea buscándola o yendo hasta ella). En la hoja Información general de la máquina virtual, en la esquina superior izquierda, haz clic en el botón Conectar (consulte la figura 6-6). Al hacer clic en ella, la hoja muestra la dirección IP que se debe usar para conectar con la máquina virtual, como se puede ver en la figura 6-6. Copia la dirección IP.

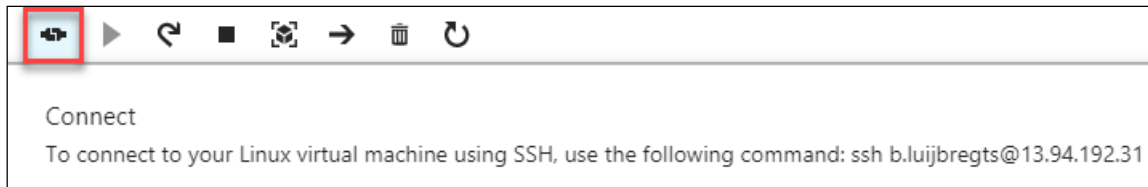


Figura 6-6: conectar a una máquina virtual en Azure Portal

Ahora puedes usar este comando directamente en Cloud Shell: solo tienes que pegarlo y pulsar Entrar. Verás una advertencia que indica que no hay ninguna clave de autenticación. Esto se debe a que elegimos trabajar con una contraseña. Elige Sí, escribe su contraseña y estarás dentro. Ahora estás conectado a Red Hat Linux Enterprise y puedes empezar a usarlo. La figura 6-7 muestra que hemos iniciado sesión en la máquina virtual a través de Cloud Shell y estamos explorando el sistema de archivos:

```
Bash | [Power] [?] [Settings]
Requesting a Cloud Shell. Succeeded.
Connecting terminal...

barry@Azure:~$ ssh b.luijbregts@13.95.29.180
Password:
Last login: Thu Sep 28 07:00:23 2017 from 52.170.90.72
[b.luijbregts@myredhatserver ~]$ ls -al /etc
total 1132
drwxr-xr-x. 84 root root    8192 Sep 28 06:54 .
dr-xr-xr-x. 17 root root    224 Aug  9 23:05 ..
drwxr-xr-x.  3 root root    101 Aug  9 23:01 abrt
-rw-r--r--.  1 root root     16 Aug  9 23:03 adjtime
-rw-r--r--.  1 root root   1518 Jun  7  2013 aliases
drwxr-xr-x.  2 root root     69 Aug  9 23:05 alternatives
-rw-----.  1 root root    541 Mar 29 16:34 anacrontab
-rw-r--r--.  1 root root     55 Mar  1  2017 asound.conf
-rw-r--r--.  1 root root      1 May 23  2016 at.deny
```

Figura 6-7: exploración del sistema de archivos de Linux utilizando Cloud Shell en Azure Portal



## Agregar almacenamiento adicional a la máquina virtual

Si deseas usar tu máquina virtual como servidor de base de datos, puedes necesitar espacio de almacenamiento extra para gestionar los datos de la base de datos. Puedes crear espacio de almacenamiento extra agregando una unidad adicional a través de Azure Portal.

1. Ve a la máquina virtual, selecciona la pestaña Discos y haz clic en Agregar disco de datos, como se muestra en la figura 6-8.

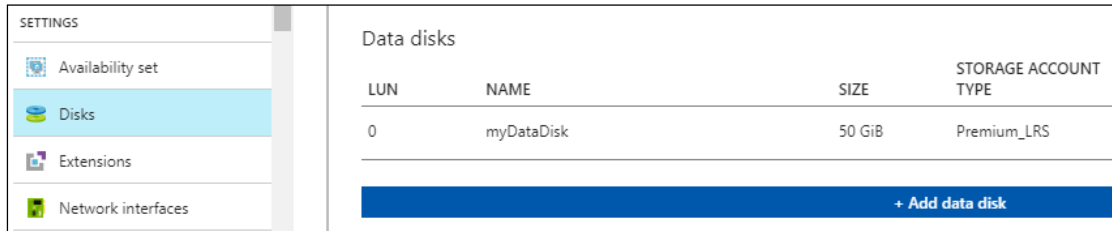


Figura 6-8: pestaña Disco de una máquina virtual en Azure Portal

Como no tenemos ningún disco que se pueda adjuntar, selecciona Crear disco.

2. Se abre el asistente Crear disco administrado. Usa este asistente para crear una unidad administrada (en lugar de una unidad no administrada).

Los discos administrados son totalmente administrados por Azure; no tienes que crear y mantener una cuenta de almacenamiento subyacente para ellos. Opcionalmente, puedes elegir el tamaño de la unidad y lo que debe haber en ella, pero vamos a dejarlo todo como está. Escribe un nombre y luego haz clic en Crear. Ahora dispones de una unidad adicional conectada a la máquina virtual.

También puedes crear y conectar una nueva unidad mediante Cloud Shell o la CLI de Azure con este comando:

```
az vm disk attach -g myResourceGroup --vm-name myVM --disk myDataDisk --new --size-gb 50
```

Si necesita ayuda con la CLI de Azure, puede escribir **az -h**. Todos los comandos de la CLI de Azure tienen la misma estructura: sustantivos az (p. ej., vm) verbo (p. ej., attach) opciones.

Has creado una nueva unidad para la máquina virtual y tienes que inicializarla en Linux. [Este documento](#) te guiará por el proceso.

Crear unidades adicionales para obtener más almacenamiento de máquina virtual es sencillo. Los pasos posteriores pueden ser más complicados, dependiendo el sistema operativo utilizado.

## Limpiar los recursos del tutorial

Si has terminado de usar la máquina virtual, puedes cerrarla y quitarla eliminando el grupo de recursos que creamos al principio del proceso para crear la máquina virtual. Este grupo contiene la máquina virtual y todos los demás recursos que se crean automáticamente. Después de eliminar el grupo de recursos, ya no pagarás por ninguno de los recursos que has utilizado en este tutorial.

## Tutorial n. 2: desarrollar una aplicación web de Node.js con MongoDB en Azure

En este tutorial, vamos a implementar una aplicación sencilla de Node.js que conecta con MongoDB y vamos a hospedarla en Azure. Hospedaremos la aplicación web en Aplicación web de Azure y utilizaremos Azure Cosmos DB como base de datos, ya que tiene una interfaz de MongoDB.

Para continuar, necesitarás tener [CLI de Azure \(v2.0 o posterior\)](#), [Git \(v2 o posterior\)](#), [Node.js y NPM \(v6 o posterior\)](#) y [MongoDB \(3.4 o posterior\)](#) instalados en el equipo.

Vamos a utilizar una aplicación de ejemplo que usa el marco de trabajo web [MEAN.js](#).

### Crear la aplicación web y la base de datos con Azure Portal

Para hospedar la aplicación de Node.js, vamos a crear una nueva aplicación web en Aplicación web de Azure.

1. En Azure Portal, haz clic en el botón Crear un nuevo servicio.
2. Busca Aplicación web. Se abre la hoja de Aplicación web. Haz clic en Crear para comenzar.
3. Se abre la hoja de creación de aplicación web.
  - a. Escribe un nombre para la aplicación web.
  - b. Crea un nuevo grupo de recursos asignándole un nombre.
  - c. Selecciona el sistema operativo que deseas usar (Windows o Linux).
  - d. Selecciona o crea un plan de App Service y haz clic en Crear.

#### Aplicación web se ejecuta con planes de App Service

App Services, al igual que Aplicación web, se ejecuta con planes de Azure App Service. Un plan de App Service es una abstracción de recursos y características, como la CPU y la memoria, y se representa por medio de niveles de precios. Los planes de App Service también están vinculados a la región geográfica que se elige. Puedes, por ejemplo, ejecutar la aplicación de Aplicación web en un plan de App Service del nivel de precios S1 (consulta la figura 6-9), que tiene 1 núcleo y 1,75 GB de RAM.

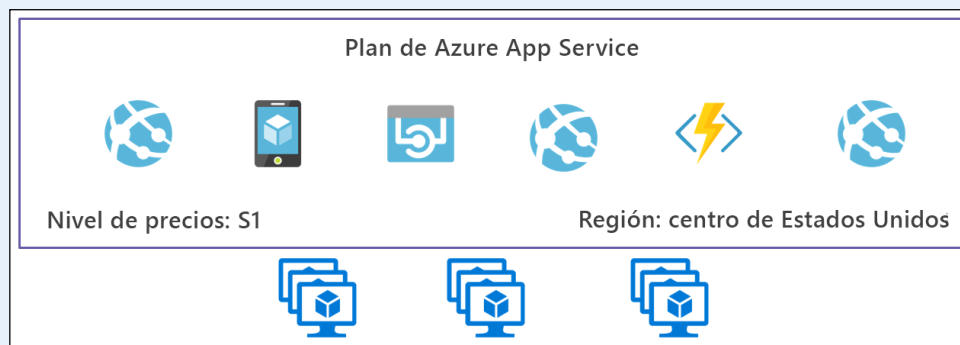


Figura 6-9: descripción de un plan de Azure App Service

Puedes ejecutar tantos App Services como desees en un plan de App Service, siempre y cuando seas consciente de que tienes que compartir los recursos entre todos los App Services.

Para hospedar la base de datos, vamos a crear una base de datos de Cosmos DB que utiliza la API de MongoDB. Funcionará exactamente del mismo modo que MongoDB y tendrá todas las ventajas de Cosmos DB.

1. En Azure Portal, haz clic en el botón Crear un nuevo servicio.
2. Busca Cosmos DB. Se abre la hoja de Cosmos DB. Haz clic en Crear para comenzar.
  - a. Escribe un nombre (identificación) para la cuenta de la base de datos.
  - b. Elige la API de MongoDB.
  - c. Selecciona el grupo de recursos que creaste para la aplicación web.
  - d. Elige la ubicación. Preferiblemente, una ubicación que coincida con la de la aplicación web o esté cerca de ella.
  - e. Haz clic en Crear.

### Ejecutar la aplicación de Node.js y la base de datos de MongoDB localmente

Vamos a ejecutar la aplicación localmente antes de ejecutarla en Azure.

Si no lo has hecho aún, [instala MongoDB Community Edition](#):

1. Abre un símbolo del sistema en su equipo y ve al directorio bin de tu instalación de MongoDB. Podría ser algo como C:\Archivos de programa\MongoDB\Server\3.4\bin.
2. Escribe **mongo** para conectar al servidor local de MongoDB.
3. Si no puedes, asegúrate de que [la base de datos de MongoDB se ha iniciado](#). En muchos casos, hay que empezar ejecutando mongod.

Ahora clona la aplicación de arranque de Node.js y ejecútala:

4. En la ventana de comandos, ve al directorio en el que desees descargar el código de la aplicación.
5. Ejecuta `git clone https://github.com/Azure-Samples/meanjs.git` para clonar la aplicación.
6. Inicie la aplicación ejecutando los siguientes comandos:

```
cd meanjs
npm install
npm start
```

7. Cuando termine de cargarse, verás un mensaje que contiene la dirección URL de la aplicación. Abre un navegador y ve a la dirección del servidor (algo similar a `http://localhost:3000`). Debe tener un aspecto parecido al de la figura 6-10.

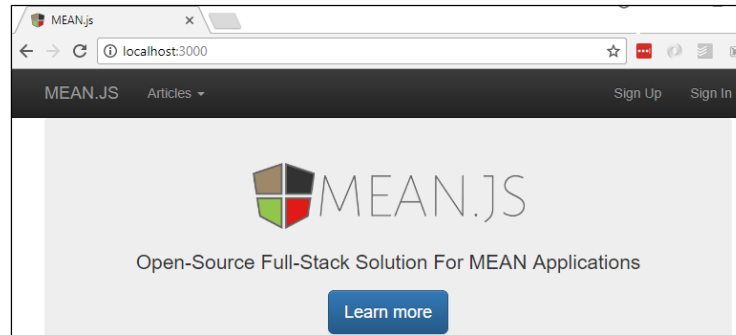


Figura 6-10: plantilla de aplicación de Mean.JS

8. En la barra de menús, haz clic en Registrarse e intenta crear un nuevo usuario. Si funciona, la aplicación tendrá acceso a la base de datos de MongoDB local y la usará para escribir los usuarios.

## Implementar la aplicación de Node.js en Azure

Ahora tienes una aplicación funcional que se ejecuta localmente. Vamos a implementarla en Azure. Para empezar, implementaremos la aplicación de Node.js en la aplicación web de Aplicación web que creamos antes.

Vamos a utilizar la CLI de Azure para transferir la aplicación de Git a Azure. Asegúrate de que la CLI 2.0 o posterior de Azure está instalada en tu equipo. Cuando termines, ejecuta el comando `az login` para iniciar sesión en tu suscripción de Azure. Al hacerlo, se abre un sitio web donde puedes autenticar y autorizar el uso de tu suscripción de Azure por la CLI.

Para FTP y Git local, debes tener configurado un usuario de implementación en el servidor con el fin de autenticar la implementación. Este usuario de implementación tiene nivel de cuenta y es distinto de tu cuenta de suscripción de Azure. Solo tienes que configurar este usuario de implementación una vez. En la CLI de Azure, ejecuta el siguiente comando, donde **nombre de usuario** y **contraseña** son valores que defines y rellenas tú mismo (no incluyas los caracteres `<` y `>`):

```
az webapp deployment user set --user-name <nombre de usuario> --password <contraseña>
```

Luego, asociamos el repositorio Git local con la aplicación web ejecutando lo siguiente:

```
az webapp deployment source config-local-git --name <nombre_aplicación> --resource-group <nombre_grupo_recurso>
```

Esto devolverá una dirección URL con el formato:

```
https://<nombre_usuario>@<nombre_aplicación>.scm.azurewebsites.net:443/<nombre_aplicación>.git
```

Ahora usaremos esta dirección URL para transferir la aplicación a Azure. En primer lugar, agregamos un Azure remoto al repositorio Git local. Ejecuta `cd` para cambiar al directorio `meanjs` del código fuente de la aplicación. Ahora ejecuta:

```
git remote add azure <url_de_implementación>
```

Para implementar la aplicación, ejecuta `git push azure master`. Se te pedirán las credenciales que creaste antes. La implementación puede tardar un tiempo en completarse. Recibirás una notificación cuando termine.

Una vez completada, podrás ir a la aplicación usando `http://<nombre_aplicación>.azurewebsites.net`.

Ahora, la aplicación se implementa en la aplicación web y tenemos una base de datos de Cosmos DB que se ejecuta con la API de MongoDB, pero todavía no están conectadas. Es lo que vamos a hacer ahora.

## Conectar la aplicación y la base de datos

Para conectar la aplicación a la base de datos, necesitas saber dónde está la base de datos. Podemos hacerlo definiendo la variable de entorno MONGODB\_URI en la aplicación web. Podemos hacerlo a través de la CLI de Azure ejecutando el siguiente comando:

```
az webapp config appsettings update \  
--name <nombre_aplicación> \  
--resource-group myResourceGroup \  
--settings  
MONGODB_URI="mongodb://<cosmosdb_name>:<clave_maestra_principal>@<nombre_cosmosdb>.documents.azure.  
com:10250/mean?ssl=true"
```

Puedes encontrar la cadena de conexión Cosmos DB que necesitas utilizar para el valor MONGODB\_URI en Azure Portal. La encontrarás en el menú Cadena de conexión de la cuenta de Cosmos DB.

Después de ejecutar el comando con la CLI de Azure, la configuración de la aplicación web tendrá un aspecto similar al de la figura 6-11.

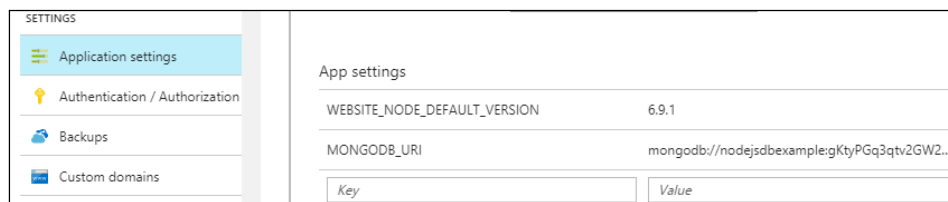


Figura 6-11: configuración de la aplicación de Aplicación web de Azure

En lugar de ejecutar el comando de la CLI de Azure, podrías haber agregado también el ajuste a través de Azure Portal.

Ahora, ve de nuevo a la aplicación web utilizando `http://<nombre_aplicación>.azurewebsites.net`. Intenta registrarte como un nuevo usuario. Cuando lo hagas, la aplicación se comunicará con la base de datos. Ahora tienes una aplicación de Node.js con Azure Cosmos DB que usa la API de MongoDB.

## Configuración de entrega continua con GitHub

Hasta ahora hemos estado enviando código desde nuestro repositorio local de Git a Azure. Esto está bien si se trabaja solo, pero si se trabaja en un equipo se necesita otro tipo de control del código fuente, como Visual Studio Team Services o GitHub.

Vamos a usar GitHub para transferir nuestro código y enlazarlo a nuestra aplicación web para que los cambios se implementen automáticamente en un proceso de entrega continua (CD).

1. Ve a <https://github.com/> y crea un nuevo repositorio.  
Debes iniciar sesión o crear una nueva cuenta para hacerlo.
2. Crea un nuevo repositorio yendo a <https://github.com/new>.
  - a. Escribe un nombre para el repositorio.
  - b. Dejar los otros ajustes como están (repositorio público, no crear un archivo README)
  - c. Crea el repositorio. Al hacerlo, aparece una pantalla similar a la figura 6-12.

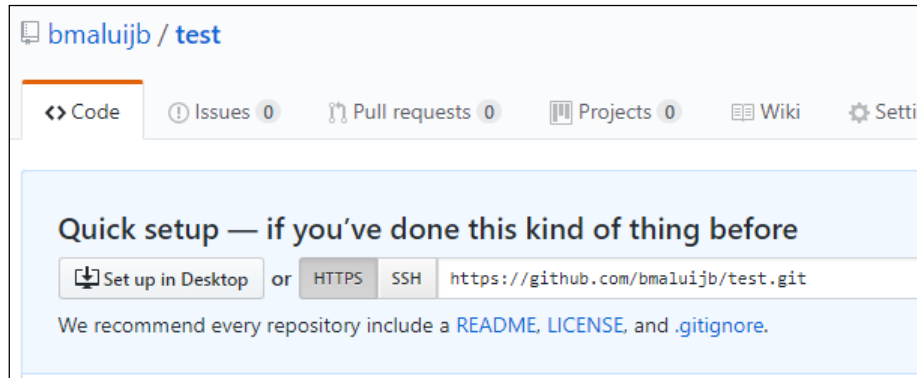


Figura 6-12: enlace del repositorio en GitHub

3. Usa la dirección URL para establecer un destino remoto para el repositorio local de Git. Puedes hacerlo en la ventana de comandos.
  - a. Ejecuta `cd` para cambiar al directorio `meanjs` del código fuente de la aplicación.
  - b. Ejecuta el siguiente comando:

```
git remote add github https://github.com/bmaluijb/test.git
```
  - c. Ejecuta el siguiente comando para insertar el código en GitHub:

```
git push github
```

Ahora, nuestro código está en GitHub y se puede compartir con el equipo. A continuación, vamos a configurar CD. Usaremos la característica Opciones de implementación de Aplicación web a través de Azure Portal. También podríamos haber usado la característica Entrega continua de Aplicación web, pero eso requiere una cuenta de Visual Studio Team Services y hemos preferido simplificar el proceso.

1. En Azure Portal, ve a la aplicación web que hospeda la aplicación de Node.js.
2. En la barra de menús, haz clic en Opciones de implementación.
3. Es posible que esto ya esté configurado para el repositorio local de Git. Si es así, debes hacer clic en Desconectar.
4. En Elegir origen, selecciona GitHub.
5. En la sección Autorización, autoriza a Azure para usar GitHub haciendo clic en Autorizar y otorgando el permiso.
6. En la sección Elegir proyecto, elige el repositorio de GitHub que acabas de crear.
7. Deja Rama definida como maestra.
8. Haz clic en Aceptar y espera unos segundos.

Vuelve al menú Opciones de implementación. Ahora puedes ver que está conectado. A partir de ahora, cada vez que insertes una nueva versión de código fuente en GitHub, se compilará e implementará en la aplicación web automáticamente, como se muestra en la figura 6-13. También puedes forzar este proceso haciendo clic en el botón Sincronizar.

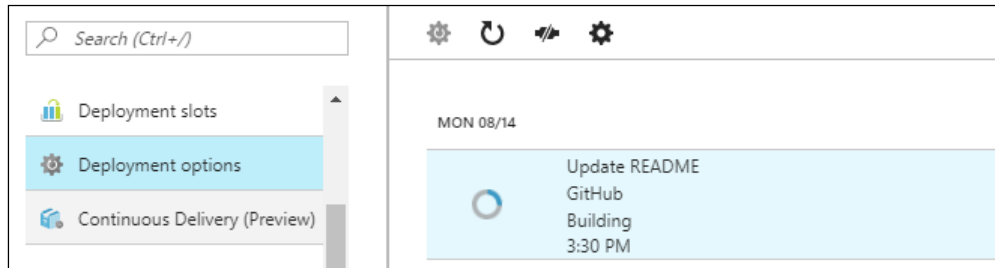


Figura 6-13: hoja Opciones de implementación de Aplicación web

## Configuración de entornos de ensayo

Con Aplicación web, puedes configurar una ranura de ensayo para probar en ella la nueva versión de tu aplicación. Puedes usar las [ranuras de implementación](#) para esto. Las ranuras de implementación son App Services completos que puede utilizar para probar el código antes de promoverlo a la siguiente ranura.

Puedes tener ranuras de implementación para el ensayo, las pruebas de carga y la producción (que es siempre el App Service original; en nuestro caso, la aplicación web de Node.js). De hecho, puedes tener tantas ranuras de implementación como desees sin incurrir en costes adicionales. Todas las ranuras de implementación se ejecutan en el mismo plan de App Service, y eso es lo que se paga. Debes recordar que tener ranuras de implementación adicionales en un plan de App Service consumirá recursos como CPU y memoria.

Puedes crear nuevas ranuras de implementación desde el elemento de menú Ranuras de implementación de la aplicación web. Asegúrate de que estás ejecutando la aplicación web en el nivel de precios Estándar o Premium, ya que el plan gratuito no incluye ranuras de implementación.

En cada ranura de implementación que crees, puedes configurar las opciones de implementación como lo hicimos antes para implementar el código automáticamente. Puedes incluso trabajar en diferentes *ramas de código fuente* para distintos entornos e implementar automáticamente ramas concretas en ranuras de implementación específicas.

Además, puedes probar la versión final en una ranura de implementación y, cuando estés satisfecho, *intercambiarla* con la ranura de producción. Esto prepara la aplicación antes del intercambio, lo que se traduce en una implementación sin tiempo de inactividad.

Veamos el proceso de creación de una ranura de implementación y el cambio a ella.

1. En Azure Portal, ve a la aplicación web que hospeda la aplicación de Node.js.
2. En la barra de menús, haz clic en Ranuras de implementación. Se abre la hoja Ranuras de implementación.
3. Haz clic en el signo más (+) para crear una nueva ranura de implementación.
  - a. Escribe un nombre para la ranura (p. ej., ensayo).
  - b. Elige la aplicación web de Node.js como Origen de configuración (esto copia la configuración de la aplicación en la nueva ranura).
  - c. Haz clic en Aceptar para crear la ranura.
4. Una vez creada la ranura, es similar a la aplicación web original.
  - a. Configura CD para la ranura como lo hiciste en el procedimiento anterior para la aplicación web.

- b. Desconecta la conexión de CD en la aplicación web original de Node.js. Así, cuando se inserta nuevo código, se entrega solamente en la ranura de ensayo.

5. Haz un cambio en la aplicación de Node.js.

- a. Cambia el texto del archivo `home.client.view.html` (puedes encontrarlo en la carpeta `meanjs\modules\core\client\views`).
- b. Confírmalo en Git e insértalo en GitHub, como al implementar la aplicación de Node.js.

La nueva versión está ya en la ranura de ensayo y no en la aplicación web original (lo que llamamos ranura de producción). Puedes comprobarlo yendo a la dirección URL de la aplicación web de Node.js y a la dirección URL de la ranura de ensayo (que puedes encontrar en la hoja Información general de la ranura, al igual que en la hoja Información general de la aplicación web).

Vamos a poner la nueva versión en producción.

1. En Azure Portal, vaya a la aplicación web de Node.js.
2. En la barra de menús, haz clic en Ranuras de implementación para abrir la hoja Ranuras de implementación.
3. Haz clic en el botón Intercambiar para abrir la hoja Intercambiar.
  - a. Deja todos los ajustes como están.
  - b. Haz clic en Aceptar para iniciar el intercambio.

Una vez finalizado el intercambio, la nueva versión de la aplicación web de Node.js estará en producción (pruébala yendo a la dirección URL de la aplicación web de Node.js). Usar las ranuras de implementación de este modo es muy beneficioso porque permite probar la nueva versión antes de que entre en producción e implementarla después en producción sin tiempo de inactividad.

## Diagnósticos adicionales con registros de diagnóstico

Cuando se ejecuta la aplicación, es vital saber cómo está funcionando. Una forma excelente de supervisar la aplicación es usar registros de diagnóstico para ver el registro de diagnóstico en directo de la aplicación web. Puedes incluso insertarlos en la ventana de la consola. Para ello, ejecuta el siguiente comando en la CLI de Azure:

```
az webapp log tail --name <nombre_aplicación> --resource-group myResourceGroup
```

Verás un registro cuando utilices la aplicación en la aplicación web para generar tráfico.

## Cómo configurar una supervisión avanzada y alertas

Otra forma excelente de hacer un seguimiento del rendimiento de tu aplicación es usar [Application Insights](#). Es una herramienta de supervisión de Azure que puede mantenerte informado del funcionamiento de tu aplicación, desde cuántos visitantes la han usado al número de excepciones que se han producido y el punto del código en el que se han registrado. A diferencia de los Registros de diagnóstico, Application Insights tiene asociada una tarifa nominal.

Puedes configurar Application Insights en Azure Portal desde la aplicación web.

1. Ve a Azure Portal y a la aplicación web que hospeda la aplicación de Node.js.
2. En la barra de menús, haz clic en Application Insights.
3. Selecciona Crear recurso nuevo.
  - a. Escribe un nombre y selecciona una ubicación para la instancia de Application Insights.



- b. Haz clic en Aceptar. Ahora, Application Insights se implementará y empezará a recopilar datos para la aplicación.

Es necesario configurar la aplicación para que empiece a enviar datos a Application Insights. Para nuestra aplicación de ejemplo de Node.js, tendrás que hacer lo siguiente:

1. En el símbolo del sistema, ejecuta `cd` para ir a la carpeta `meanjs` de la aplicación de Node.js.
2. Ejecuta `npm install applicationinsights --save`.
3. Ve a Azure Portal y recupera la clave de instrumentación de Application Insights.  
La encontrarás en la instancia de Application Insights, en Propiedades.

4. Abre el archivo `server.js` de la aplicación web de Node.js.
5. Agrega las tres líneas siguientes de código JavaScript *antes de* la línea:

```
var app = require('./config/lib/app');  
const appInsights = require("applicationinsights");  
appInsights.setup("<clave_instrumentación>");  
appInsights.start();
```

6. Inserta los cambios en GitHub e inicia la implementación automática. En el símbolo del sistema, ejecuta lo siguiente:  

```
git commit -a -m "added Application Insights support"
```
7. Ejecuta `git push github` y los cambios se insertarán en GitHub y se implementarán automáticamente en la aplicación web.

Una vez terminada la implementación, la aplicación enviará datos a Application Insights.

De manera predeterminada, Application Insights realiza *detección inteligente*. Esta es una característica muy eficaz que detecta si algo va mal y alerta al usuario. Puede detectar cosas como un aumento repentino de las solicitudes con error y si la aplicación funciona con una lentitud inusual. También puedes crear tus propios eventos personalizados para todo tipo de métricas y condiciones en el menú Alertas de Application Insights, como se muestra en la figura 6-14.

	NAME	CONDITION	LAST ACTIVE
Previews	MYGITHUBTEST (COMPONENTS)		
Properties			
Alerts	Failed requestspers second	Failed Requests per Second > 1	Never

Figura 6-14: alertas de Application Insight

Ahora, ve a Azure Portal, busca el recurso de Application Insights y haz clic en él. Allí verás la información general, que muestra métricas básicas como el tiempo de respuesta del servidor, el tiempo de carga de visualización de las páginas y el número de solicitudes del servidor y de solicitudes fallidas. Si ves datos aquí, significa que Application Insights está funcionando.



**Nota** Puedes recopilar información adicional para enviarla a Application Insights siguiendo [esta guía](#).

## Escalado de la aplicación web

Cuando se tienen muchos usuarios, es necesario que la aplicación web ajuste su escala para dar cabida al tráfico. Cuando no está ocupada, conviene que la escala se reduzca de nuevo para ahorrar costes. Puedes hacerlo usando la función de escalado automático de App Service. Ten en cuenta que debes ejecutar la aplicación web en el nivel de precios Estándar o Premium para utilizar esta función.

La aplicación web tiene un elemento de menú denominado Escalar horizontalmente, como se muestra en la figura 6-15. Puedes usarlo para ampliar la escala manual o automáticamente.

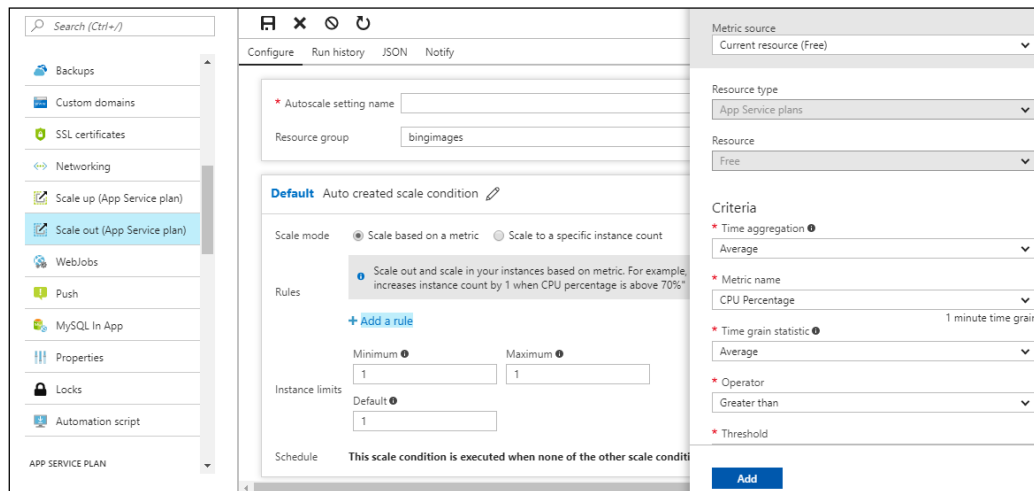


Figura 6-15: escalar una aplicación web a través de Azure Portal

Escalar horizontalmente significa agregar más instancias de la aplicación para gestionar la carga. Cuando se escala o se reduce horizontalmente de forma automática, se puede hacer en función de determinadas métricas, como el porcentaje de uso de la CPU o la memoria, o según una programación (todos los días a las 17:00) o con una mezcla de ambos. Es muy fácil de configurar y supervisar.

## ¡Listo para la producción! Agregar Capa de sockets seguros

Ahora que la aplicación está lista para la producción, debes confirmar que es segura. Además de la autenticación y la autorización, suministrar la aplicación web a través de HTTPS es una de las cosas más importantes que puedes hacer. El motivo es que, sin HTTPS, los intrusos podrían ver el tráfico entre tus recursos y utilizarlo con fines maliciosos (como iniciar sesión en tu aplicación). Además, HTTPS es obligatorio para características avanzadas como los [services workers](#).

Suministrar tráfico a tu aplicación web a través de Capa de sockets seguros (SSL) es posible mediante la importación de un certificado SSL en la aplicación web y su enlace con uno de tus nombres de dominio (personalizados). Puedes importar tu propio certificado SSL, comprado por ti o por la compañía, o comprar uno nuevo a través de [Certificados de Azure App Service](#), que facilita la compra y la validación del certificado. Después de importar el certificado, debes asociarlo a uno de los enlaces de nombres de dominio de tu aplicación web. Puedes hacer todo esto desde el menú Certificados SSL de la aplicación web.

## Tutorial n. 3: crear un back-end para una aplicación móvil con Azure

Hoy en día, casi todo el mundo tiene un dispositivo móvil. El tráfico de la mayoría de las aplicaciones procede de dispositivos móviles y esa tendencia sigue creciendo. Crear una aplicación móvil para tu compañía es una cosa, pero proporcionar una experiencia sólida y fiable es muy distinto. Azure puede ayudarte a hacerlo por medio de un back-end avanzado con [Aplicación móvil de Azure App Service](#).

Las aplicaciones de Aplicación móvil de Azure constan de dos partes:

- Un back-end que proporciona API para comunicarse con una base de datos y realizar acciones como la autenticación, la sincronización sin conexión y las notificaciones push
- SDK para todo tipo de plataformas que permiten conectar fácilmente con el back-end

Puedes crear el back-end de la aplicación de Aplicación móvil en Node.js y C#. El SDK cliente está disponible para prácticamente todas las plataformas móviles en uso. Para aplicaciones móviles multiplataforma, nativas e híbridas. Está disponible para Android, Cordova, iOS, Windows, Xamarin.Android, Xamarin.Forms y Xamarin.iOS.

En este tutorial, creamos con C# un nuevo back-end de aplicación móvil que utilizamos con una aplicación Android creada con Xamarin.Android.

Para seguir, es necesario instalar la [carga de trabajo Desarrollo para dispositivos móviles con .NET](#) con el instalador de [Visual Studio 2017](#). Esto instala las herramientas de Xamarin para Android y los SDK necesarios.

Si trabajas en un Mac, puedes [hacer lo mismo](#) en Visual Studio para Mac.

### Crear una aplicación móvil a través de Azure Portal

Empezar a trabajar es muy fácil. Vamos a crear el back-end móvil y una aplicación de arranque para el cliente a través de Azure Portal.

1. En Azure Portal, haz clic en Crear un nuevo servicio y, en el cuadro Buscar, escribe **Mobile App Quickstart**.
2. Selecciona el resultado Mobile App Quickstart y haz clic en Crear.

Se abre el asistente de creación de aplicaciones móviles.

- a. Escribe un nombre para tu aplicación móvil.
- b. Crea un nuevo grupo de recursos.
- c. Crea un nuevo plan de App Service para ejecutar la aplicación móvil. Puedes usar el nivel de precios Gratis para este tutorial.
- d. Deja el ajuste de Application Insights desactivado.
- e. Haz clic en Crear. Ya está. Ahora se implementará la aplicación móvil.

Una vez que la aplicación móvil se implemente, puedes empezar a utilizarla. En Azure Portal, en la aplicación móvil, haz clic en el elemento de menú Inicio rápido y selecciona Xamarin.Android. Se abre el asistente de inicio rápido.

### Paso 1: conectar a una base de datos

Tienes que conectar una base de datos para almacenar los datos. Haz clic en el icono de información para empezar. En la pestaña Conexiones de datos que acabas de abrir, haz clic en Crear un nuevo servicio. En el asistente Agregar la conexión de datos que se abre, haz lo siguiente:

1. En Tipo, selecciona Base de datos SQL.
2. Crea una nueva base de datos con un nuevo servidor de destino.
3. Deja la cadena de conexión como está.
4. Haz clic en Aceptar.

Se crea la conexión de datos.

### Paso 2: crear una API de tabla

Para comunicarse con la base de datos, necesitas una API. Puedes crear un back-end de API hospedado en tu aplicación móvil siguiendo estos pasos:

1. En el área Paso 2, selecciona C# y después haz clic en Descargar. Esto descarga el proyecto de back-end que se va a implementar en la aplicación móvil.
2. Descomprime el archivo que acabas de descargar y ábrelo con Visual Studio 2017 o Visual Studio para Mac.
3. En Visual Studio, haz clic con el botón secundario en el archivo de proyecto y selecciona Publicar.
4. Selecciona App Service, selecciona Existente y haz clic en Publicar.
5. Busca tu aplicación móvil y luego haz clic en Aceptar para comenzar a publicar. Ya está. El back-end móvil está funcionando.

Deja Paso 3 como **Crear una aplicación nueva** y haz clic en **Descargar**.

Descomprime el archivo que acabas de descargar y ábrelo en Visual Studio 2017 o Visual Studio para Mac.

Ejecuta la aplicación pulsando F5. Esto inicia el emulador de Android e implementa la aplicación cliente. La aplicación cliente es una aplicación de tareas pendientes. Se agrega un elemento en el cuadro de texto y se escribe en el back-end de tu aplicación móvil.

### Agregar autenticación a la aplicación

Ahora que tienes una aplicación móvil funcional con un back-end, debes asegurarte de que está protegida. Vamos a agregar autenticación a través de Azure Active Directory. Para completar este proceso, primero tendrás que registrar la aplicación móvil para la autenticación con Azure Active Directory. ([Este tutorial](#) muestra cómo hacerlo).

### Proteger la aplicación

En las API del back-end, debes definir cuáles deseas proteger.

1. Ve a tu proyecto de back-end de aplicación móvil en Visual Studio 2017 o Visual Studio para Mac.
2. Ve a Controladores > TodoItemController.cs.
3. Agrega el atributo [Authorize] a la clase TodoItemController.  
Esto restringe cualquier uso sin autenticar de este controlador.
4. Haz clic con el botón secundario en el archivo de proyecto y luego publícalo en tu aplicación móvil.

### Agregar autenticación a la aplicación móvil

El back-end está protegido. Ahora tienes que notificárselo a la aplicación móvil para que pueda iniciar sesión. Tenemos que exigir a los usuarios que se autenticuen antes de usar la aplicación.

1. En Visual Studio, ve al proyecto de aplicación móvil de Xamarin.
2. Agrega el código siguiente a la clase TodoActivity:

```
// Define a authenticated user.
private MobileServiceUser user;
private async Task<bool> Authenticate()
{
    var success = false;
    try
    {
        // Sign in with AAD login using a server-managed flow.
        user = await client.LoginAsync(this,
            MobileServiceAuthenticationProvider.WindowsAzureActiveDirectory);
        CreateAndShowDialog(string.Format("Ha iniciado sesión - {0}",
            user.UserId), "Sesión iniciada.");

        success = true;
    }
    catch (Exception ex)
    {
        CreateAndShowDialog(ex, "Error de autenticación");
    }
    return success;
}

[Java.Interop.Export()]
public async void LoginUser(View view)
{
    // Load data only after authentication succeeds.
    if (await Authenticate())
    {
        //Hide the button after authentication succeeds.
        FindViewById<Button>(Resource.Id.buttonLoginUser).Visibility = ViewStates.Gone;

        // Load the data.
        OnRefreshItemsSelected();
    }
}
```

Esto crea un nuevo método para autenticar un usuario y un controlador de método para un nuevo botón Iniciar sesión.

3. En el método OnCreate, elimina la siguiente línea de código o desactívala marcándola como comentario:  
OnRefreshItemsSelected();

4. En el archivo Activity\_To\_Do.xml, agrega la siguiente definición de botón LoginUser antes del botón AddItem actual:

```
<Button
    android:id="@+id/buttonLoginUser"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:onClick="LoginUser"
    android:text="@string/login_button_text" />
```

5. Agrega el siguiente elemento al archivo de recursos Strings.xml:

```
<string name="login_button_text">Iniciar sesión</string>
```

6. Ejecuta la aplicación.

Ahora aparece un botón Iniciar sesión junto al botón Agregar. No podrás agregar ningún elemento sin iniciar sesión porque todas las acciones de `ToDoItemController` son ahora seguras. Al hacer clic en Iniciar sesión, se dirige al usuario a la pantalla de autenticación de Azure Active Directory. En ella, usa la cuenta que utilizas para iniciar sesión en Azure Portal. Esta cuenta se agrega a Azure Active Directory de manera predeterminada. Inicia sesión y verás que se inicia correctamente, como se muestra en la figura 6-16. Ya puedes empezar a agregar elementos de tareas pendientes.

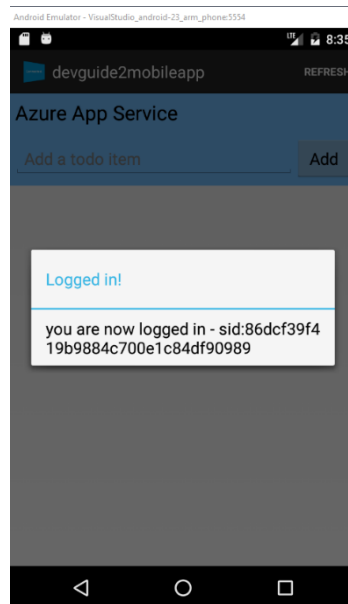


Figura 6-16: recién autenticado en la aplicación móvil en el emulador

## Agregar sincronización sin conexión a la aplicación

En muchas situaciones de uso móvil, los usuarios no siempre tienen acceso a Internet. Aunque no lo tengan, deben poder seguir trabajando. Para que esto sea posible, pueden utilizar la característica Sincronización sin conexión de Aplicación móvil de Azure. Vamos a activarla para nuestra aplicación:

1. En Visual Studio 2017 o Visual Studio para Mac, ve a la aplicación cliente de Xamarin Android.
2. En la clase `ToDoActivity`, elimina la marca de comentario de la siguiente línea:

```
#define OFFLINE_SYNC_ENABLED
```

3. Ya está. Ejecuta la aplicación para implementarla en el emulador de Android.

Ahora la aplicación funciona exactamente igual que antes, pero con algunos cambios. Usa un objeto `IMobileServiceSyncTable` en lugar de un objeto `IMobileServiceTable`. También usa el paquete NuGet de SQLite. Esto inicia una base de datos SQLite en el dispositivo Android que la aplicación utiliza para guardar los datos cuando no está online. Cuando se restaura el acceso a Internet, los cambios de la base de datos SQLite local se insertan en la aplicación móvil y los posibles cambios que se hayan producido en el back-end se sincronizan con la aplicación en el dispositivo Android. Vamos a intentarlo:

1. Ejecuta la aplicación en el emulador de Android.
2. Inicia sesión como lo hiciste antes.
3. Agrega un elemento.
4. Desconéctate de Internet. En el emulador de Android, puedes hacerlo en Extended Controls, definiendo Signal Strength of the Network como None.
5. Agrega otro elemento y completa un elemento activándolo. La aplicación sigue funcionando en la base de datos SQLite local.

Puedes comprobar que los elementos no se han escrito en la base de datos del back-end conectando a la base de datos SQL de Azure que hemos creado antes con una herramienta como [SQL Server Management Studio](#). Puedes encontrar la `connectionstring` para la base de datos SQL de Azure al ir hasta ella en Azure Portal.

6. Conecta el dispositivo Android a Internet de nuevo. En el emulador de Android, ve a Extended Controls y define Signal Strength como Great.
7. En tu aplicación móvil, haz clic en el botón Actualizar. Los cambios locales se sincronizarán ahora con Aplicación móvil de Azure.

Comprueba la base de datos SQL de Azure para ver los cambios que ahora están sincronizados con el back-end.

## Características adicionales y paso a la producción

Es fácil configurar una aplicación usando Aplicación móvil e incluir características avanzadas como la autenticación y la sincronización sin conexión.

Como la aplicación móvil es un servicio de aplicación de Azure, tiene las mismas características que el resto de los App Services. Esto significa que la aplicación móvil también puede escalarse automáticamente, que cuenta con las mismas opciones de supervisión y alertas y que puedes hacer diagnósticos adicionales con los registros de streaming e incluso incorporar depuración de aplicaciones.

Al mover la aplicación a la producción, puedes proporcionar seguridad adicional mediante los certificados de App Service. También puedes configurar CD, como lo hicimos para la aplicación de Node.js. Y puedes usar ranuras de implementación para implementar sin tiempo de inactividad.

Configurar todos estos elementos solía ser difícil; ahora son problemas resueltos. Deja que Azure se encargue del trabajo pesado para que puedas centrarte en crear cosas que importan.

# Usar Azure Marketplace

El ecosistema Azure no solo te ayuda a crear aplicaciones dinámicas de una forma más rápida y eficaz, sino que con Azure Marketplace puedes encontrar soluciones de todo tipo que facilitarán tus iniciativas de desarrollo. Y puede resultarte útil conectar tu producto con usuarios de todo el mundo.

## ¿Cómo puede ayudarme Azure Marketplace como desarrollador?

[Azure Marketplace](#) te proporciona configuraciones listas para la ejecución de todo tipo de productos, desde Windows Server a Octopus Deploy, bases de datos de Oracle, instancias de WordPress o [servidores de Minecraft](#). Solo tienes que hacer clic en una solución, configurarla, implementarla y empezar a usarla. Marketplace está lleno de productos y soluciones de Microsoft y de partners de Microsoft. También incluye muchas soluciones de código abierto.

La ventaja de Marketplace es su facilidad de uso. Las soluciones se implementan preparadas para ejecutarse, junto con todos los componentes requeridos. A veces se trata de una máquina virtual con software instalado, una red virtual configurada con una licencia integrada instalada o una licencia que trae el usuario. A veces es un servicio que se compra, como una suscripción a la API de Mapas de Bing para empresas. Puedes usar Marketplace para probar fácilmente nuevo software y para usar soluciones de nivel empresarial. Sean cuales sean tus necesidades, probablemente podrás encontrar una solución para ellas en Marketplace.



## Soluciones de Azure

Puedes aprender de compañías que usan Azure con buenos resultados a través de [Soluciones de Azure](#). Estas soluciones muestran documentación sobre los productos, casos prácticos y prácticas recomendadas de arquitectura para casos de uso de Azure frecuentes como marketing digital, SharePoint en Azure, aplicaciones de línea de negocio y aplicaciones móviles.

Lo más valioso de las soluciones de Azure para los desarrolladores son las [arquitecturas de solución](#). Te proporcionan arquitecturas de referencia para problemas frecuentes y explican por qué deberías utilizar determinados productos en algunas situaciones.

## Resumen y próximos pasos

En esta guía, hemos analizado las capacidades que Azure puede aportar a tus aplicaciones. Con Azure, puedes hacer cosas increíbles con tus aplicaciones: reconocimiento facial y de voz, administrar tus dispositivos del Internet de las cosas en el cloud, ajustar la escala como desees y pagar por lo que usas.

Has visto que, sea cual sea el lenguaje de programación que usas o la plataforma para la que escribes aplicaciones, Azure puede ayudarte con servicios para prácticamente cualquier situación.

Esperamos que sigas usando este libro electrónico para familiarizarte un poco más con la amplia gama de servicios de Azure y determinar cuáles se ajustan mejor a tu situación.

Se acabaron los días en que tenías que escribir complicadas secuencias de código; ahora puedes aprovechar una extensa oferta de soluciones predefinidas. Libérate de trabajo para concentrarte en las cosas que importan y deja que Azure se encargue de los problemas solucionados.

## Sigue aprendiendo con una cuenta gratuita de Azure

Regístrate para obtener una [cuenta gratuita de Azure](#) y recibe:

- Un crédito de 170 euros para utilizar con cualquier producto de Azure durante 30 días
- Acceso gratuito durante 12 meses a la mayoría de nuestros productos más populares de distintas categorías, como computación, almacenamiento, redes y bases de datos
- Más de 25 productos que siempre son gratuitos

# Acerca de los autores

Michael y Barry son grandes entusiastas de Azure y puedes contactar con ellos en Twitter para hacerles llegar preguntas sobre este libro.



Michael Crump trabaja en Microsoft en la plataforma Azure y es programador, bloguero y orador internacional sobre diversos temas relacionados con el desarrollo en el cloud. Le encanta ayudar a los desarrolladores a entender los beneficios del cloud de una forma clara y sencilla.

Puedes contactar con él en Twitter [@mbcrump](https://twitter.com/mbcrump) o siguiendo su blog en <https://www.michaelcrump.net>.



Barry Luijbregts es un arquitecto de software independiente y desarrollador apasionado del cloud. También es uno de los autores de Pluralsight. Ha trabajado en muchas empresas a lo largo de los últimos diez años y le encanta compartir sus conocimientos con la comunidad. Tiene un extenso conocimiento de la oferta de Microsoft, con especial énfasis en la tecnología web y el cloud. Barry colabora con un grupo de usuarios centrado en los conocimientos tecnológicos y de software llamado .NET Zuid (South) y actualmente imparte cursos sobre los beneficios del cloud. Vive en los Países Bajos con su preciosa esposa y su hija pequeña y le encanta jugar con sus dos huskies siberianos.

Puedes contactar con Barry en Twitter [@AzureBarry](https://twitter.com/AzureBarry) y a través de su sitio web en <https://www.azurebarry.com/>.