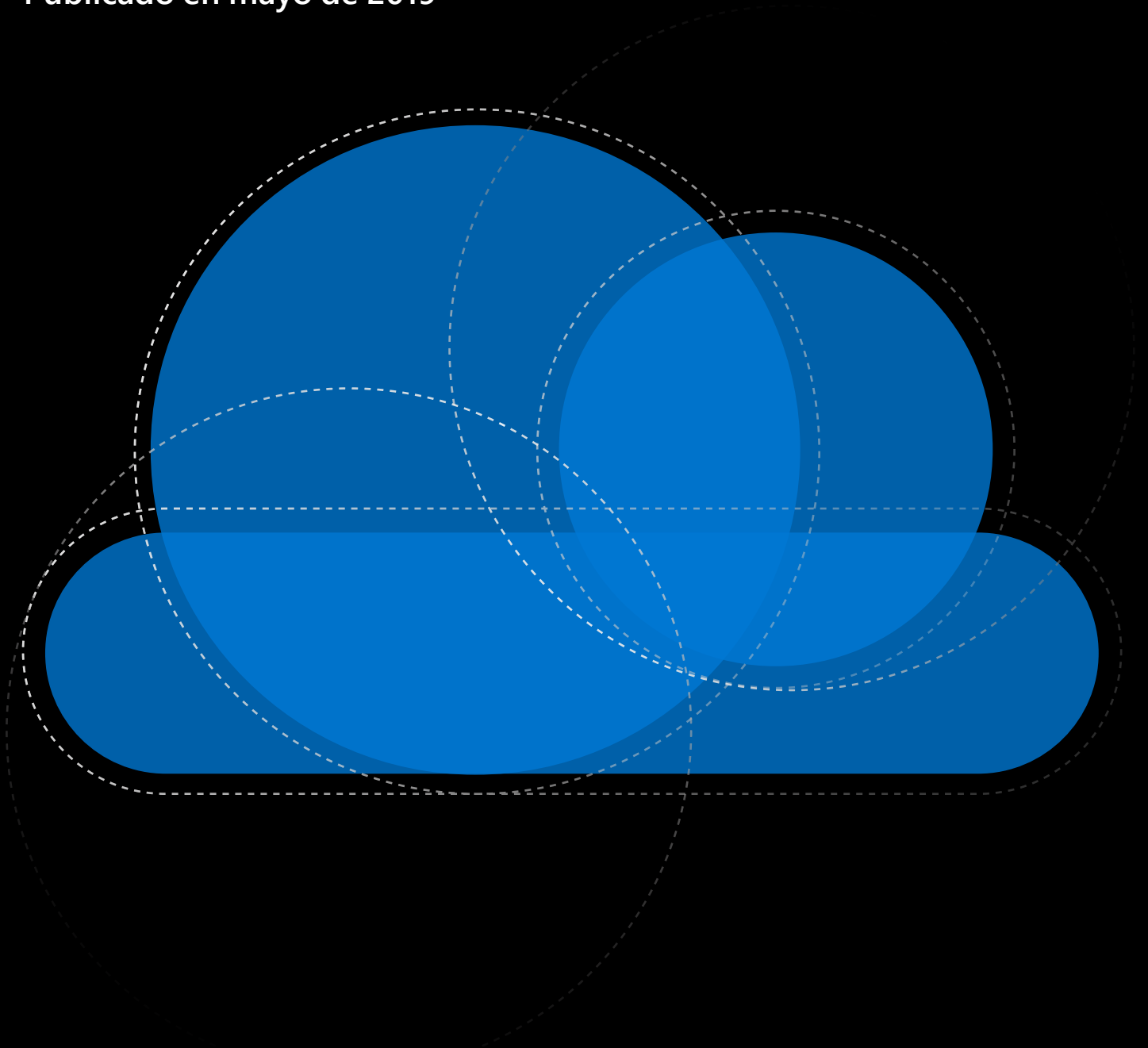


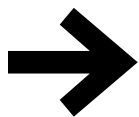
Serie de e-books



La guía de Azure para desarrolladores

Publicado en mayo de 2019





03 /

Introducción

Estamos aquí para ayudarte

05 /

Capítulo 1: Introducción a Azure

¿Qué puede hacer Azure por ti?
Dónde debes hospedar tu aplicación
Características de Azure App Service
Azure Functions
Azure Logic Apps
Azure Batch
Contenedores
Qué utilizar y cuándo
Crear aplicaciones con mejor rendimiento
Azure Front Door
Azure Content Delivery Network
Azure Redis Cache

22 /

Capítulo 2: Conectar tu aplicación con datos

¿Qué puede hacer Azure por tus datos?
Dónde almacenar los datos
Azure Cosmos DB
Azure SQL Database
Bases de datos de Azure para MySQL, PostgreSQL y MariaDB
Azure Storage
Soluciones de análisis de datos de Azure
Tutorial: Publicación de un sitio web on-premises en Azure con una base de datos SQL

40 /

Capítulo 3: Proteger una aplicación

¿Cómo puede Azure ayudar a proteger tu aplicación?
Cifrado
Azure Security Center
Registro y supervisión

51 /

Capítulo 4: Agregar inteligencia a tu aplicación

¿Cómo puede Azure integrar la IA en tu aplicación?
Búsqueda de Azure
Cognitive Services
Azure Bot Service
Azure Machine Learning Studio
Herramientas para desarrolladores de inteligencia artificial
IA y realidad mixta
Usar eventos y mensajes en una aplicación

72 /

Capítulo 5: Conecta tu empresa con IoT

¿Cómo puede Azure conectar, proteger, gestionar, supervisar y controlar tus dispositivos en el cloud?
Azure IoT Hub
Azure IoT Central
Aceleradores de soluciones de Azure IoT
Azure IoT Edge
Azure Digital Twins
Azure Sphere
Obtener más información acerca de Azure IoT
Qué utilizar y cuándo

82 /

Capítulo 6: Dónde y cómo implementar los servicios de Azure

¿Cómo puede Azure implementar tus servicios?
Infraestructura como código
Azure Blueprints
Contenedores de Azure
Azure Stack
Dónde y cuándo implementar

89 /

Capítulo 7: Compartir código, controlar el trabajo y enviar el software

¿Cómo puede ayudarte Azure a planificar de forma más sensata, colaborar mejor y enviar tus aplicaciones más rápido?
Azure Boards
Azure Repos
Azure Pipelines
Azure Test Plans
Azure Artifacts

98 /

Capítulo 8: Azure en acción

Tutorial: portal de Azure
Tutorial: Desarrollar una aplicación web y una base de datos
Tutorial: Extensión de aplicaciones
Tutorial: Listo para producción

119 /

Capítulo 9: Resumen y recursos

Sigue aprendiendo con Azure
Acerca de los autores

La guía de Azure para desarrolla- dores

Esta guía ha sido diseñada para desarrolladores y arquitectos que están comenzando su viaje hacia Microsoft Azure. En ella te explicaremos todos los entresijos de Microsoft Azure.

Aprenderás a dar los primeros pasos con la herramienta y descubrirás los servicios que podrás usar en los casos que se te presenten.

Desde la creación de sitios web, bases de datos y aplicaciones móviles y de escritorio hasta la integración de las últimas tecnologías en tu aplicación, Azure te ayuda con las tareas más pesadas. Los servicios de Azure han sido diseñados para funcionar en conjunto, por lo que podrás crear soluciones completas que aguanten durante toda la vida útil de tu aplicación.

Estamos aquí para ayudarte

Podemos ayudarte de muchas formas, según lo que necesites.

Con nuestros [planes de soporte](#) tendrás acceso a los equipos de soporte técnico de Azure, a guías de diseño en el cloud y a asistencia en la planificación de migraciones. Puedes incluso adquirir un plan de soporte que garantice las respuestas de los equipos de soporte técnico en menos de 15 minutos.

También puedes obtener ayuda a través de otros canales, tales como:

[Documentación y guías](#) que resumen la información general de todo lo que incluye Azure y proporcionan información detallada gracias a la documentación específica de cada característica.

[Acuerdos de nivel de servicio \(SLA\)](#), que pueden informarte sobre las garantías de tiempo de actividad y las políticas de crédito por tiempo de inactividad.

[@AzureSupport en Twitter](#), que es operado por hábiles ingenieros de Azure que responden con rapidez a los problemas que les presentas por Twitter.

[Stack Overflow](#), que ofrece respuestas a preguntas sobre Azure e incluye muchos mensajes activos de miembros de los equipos de ingeniería de Azure.

[La comunidad de soporte de Azure](#), que ofrece un lugar para dialogar con la comunidad de Azure y contiene respuestas a las preguntas de la comunidad.

[Azure Advisor](#), que hace recomendaciones de los recursos de Azure de forma automática, incluido lo que necesitas hacer para lograr mayor seguridad y disponibilidad, mejorar el rendimiento y reducir los costes.

[Azure Service Health](#), que te ofrece una visualización personalizada del estado de tus servicios de Azure.

01 /

Intro- ducción a Azure

Has tomado la decisión de desarrollar aplicaciones en Azure y ahora quieres ponerte a ello. No tienes que hacer demasiado: solo registrarte para obtener una [cuenta gratuita de Azure](#). La cuenta incluye créditos para explorar los servicios de pago de Azure y más de 25 servicios que podrás usar gratis siempre.

Solo tienes que elegir qué herramientas, aplicaciones y marcos quieres usar y después puedes comenzar a ejecutar tus aplicaciones en Azure.

¿Qué puede hacer Azure por ti?

Ya seas un desarrollador profesional o escribas código por diversión, Azure pone a tu disposición la última tecnología en el cloud y las mejores herramientas para desarrolladores de su clase, y facilita el desarrollo en el cloud en el lenguaje que prefieras.

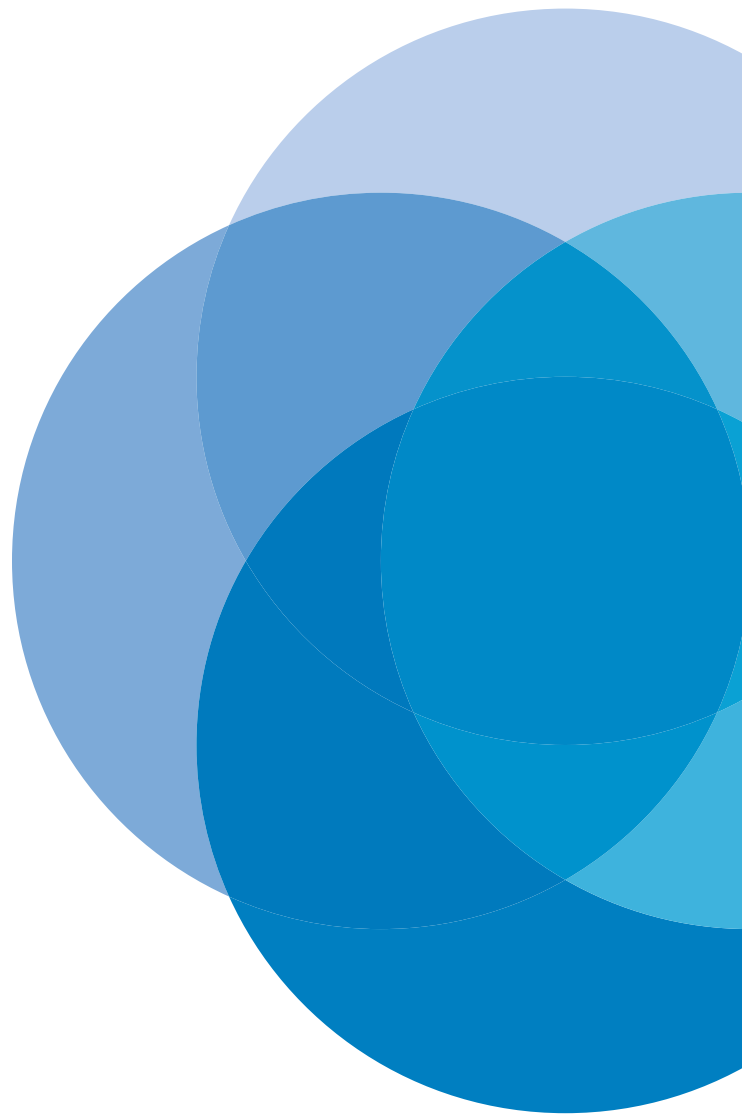
Con Azure, puedes sacar adelante tu trabajo más rápido, llevar tus capacidades al siguiente nivel e imaginar y crear las aplicaciones del mañana.

Multiplifica tu impacto con:

- Una plataforma en el cloud.
- Herramientas desarrollo.
- Servicios de administración

Todas estas características, integradas en conjunto, forman un verdadero ecosistema que te permite crear aplicaciones increíbles y experiencias digitales continuas que podrás ejecutar en cualquier dispositivo.

Aprovecha las increíbles y crecientes capacidades de Azure. Vamos a ponernos manos a la obra y veamos qué puedes hacer.



Dónde debes hospedar tu aplicación

Azure ofrece servicios que han sido diseñados para facilitarte todo lo que necesitas para desarrollar y escalar cualquier aplicación. Al utilizar los servicios de Azure para ejecutar tu aplicación, disfrutarás de escalabilidad, alta disponibilidad, una plataforma totalmente administrada y servicios de base de datos. Azure también ofrece las siguientes opciones para ejecutar tu aplicación.

Azure App Service

Puedes hospedar tus aplicaciones en una plataforma de aplicaciones totalmente administrada que a los desarrolladores empresariales les encanta: [Azure App Service](#). Azure App Service es un conjunto de servicios de hosting y orquestación con características y capacidades comunes. Todos los servicios de App Service tienen la capacidad, por ejemplo, de proteger una aplicación con [Azure Active Directory](#) y pueden usar dominios personalizados.

Azure App Service se compone de lo siguiente:

Web Apps: [Web Apps](#) es uno de los servicios de Azure más utilizados y puede 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. Web Apps puede hospedar aplicaciones que están escritas en .NET, Node.js, Python, Java o GO, y hay extensiones disponibles que se pueden usar para ejecutar incluso más lenguajes.

```
// Pruébalo: crea una aplicación  
web ASP.NET Core en Azure
```

Mobile Apps: Disfruta de un back end para tus aplicaciones móviles con [Mobile Apps](#). Al hospedar una API en Mobile Apps, tus aplicaciones móviles se conectarán a ella mediante el SDK de cliente multiplataforma. Está disponible para Windows, iOS y Android. Mobile Apps 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.

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 pone sobre la mesa capacidades exclusivas, pero todos ellos comparten algunas características comunes:

Escalado

Azure App Service se ejecuta en [planes de App Service](#), que son abstracciones de máquinas virtuales (VM). Habrá una o varias VM que ejecutarán tu Azure App Service, pero como Azure se ocupa de todo, no hace falta que sepas cuáles. Sin embargo, puedes ajustar la escala de los recursos que ejecuta 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 escale automáticamente el número de instancias por ti en función de una programación o de métricas como la CPU, la memoria o la longitud de la cola HTTP.

Ranuras de implementación

Tras implementar una nueva versión de tu aplicación en una ranura de implementación, puedes probar si funciona de la manera esperada y trasladarla 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, si desvías al 10 por ciento de tus usuarios a la nueva versión de la aplicación en la ranura de implementación, podrás ver si las nuevas características funcionan de la forma esperada y si los usuarios las están utilizando.

Cuando estés satisfecho con el rendimiento de la nueva versión de tu aplicación en la ranura de implementación, podrás realizar un "intercambio", que reemplazará la aplicación de la ranura de implementación por la aplicación de la de producción. También puedes cambiar de una ranura de desarrollo a una ranura de ensayo y luego a la ranura de producción. 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. También puedes deshacer el intercambio y revertir la implementación de la nueva versión.

Puedes usar ranuras de implementación dentro de los entornos, como los de desarrollo, prueba o producción. No utilizas ranuras de implementación como entornos, porque todas residen en el mismo plan de App Service.

Deben estar separadas por motivos de seguridad, escalado, facturación y rendimiento. 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 Azure DevOps realicen operaciones de intercambio durante un lanzamiento.

Implementación continua

Para publicar tu aplicación en App Service, puedes usar servicios externos como Jenkins, Octopus Deploy o muchos más. También puedes usar la [característica de implementación continua \(CD\) de Azure DevOps](#) en App Service. 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)

Conexión a recursos on-premises

Puedes conectarte a recursos externos, como almacenes de datos, a tus App Services. No es necesario que estos recursos se encuentren en Azure; pueden estar en cualquier sitio, por ejemplo on-premises o en tu propio centro de datos. En función de tus requisitos, podrás conectarte a servicios on-premises mediante muchos mecanismos, por ejemplo con [Azure Hybrid Connections](#), [Azure Virtual Networks](#) y [Azure ExpressRoute](#).

Dominios personalizados y certificados de Azure App Service

Al lanzar una aplicación en Azure App Service, expone una URL, por ejemplo `https://myazurewebsite.azurewebsites.net`. Es muy probable que quieras usar tu propio dominio personalizado. Para ello, tienes que asignar ese nombre de dominio a App Services. [Esta es la forma de hacerlo](#).

Además, puedes asegurarte de que la aplicación se sirve a través de HTTPS usando un certificado de capa de sockets seguros (SSL). Solo tienes que traer tu propio certificado o comprar uno [directamente desde el portal de Azure](#). Al comprar un certificado SSL en el portal de Azure, estarás comprando un certificado de Azure App Service. Puedes configurarlo para utilizarlo en las conexiones de tu dominio personalizado.

```
// Pruébalo: Compra y configura  
un certificado SSL en este  
tutorial
```

Entorno de App Service

En una aplicación web de varios niveles, a menudo se tiene una base de datos o servicios que utiliza una aplicación en Web Apps. Lo deseable es que esos servicios solo estén expuestos a la aplicación y no a Internet. Sin embargo, como supone el punto de entrada para tus usuarios, la aplicación en sí suele estar expuesta a Internet.

Para aislar estos servicios de soporte de Internet, puedes utilizar una red virtual de Azure.

Este servicio empaqueta tus servicios de soporte y los conecta a tu aplicación en Web Apps 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 puede que quieras tener aun más control. Tal vez quieres que tu aplicación esté contenida en una red virtual para poder controlar el acceso.

Quizás quieras que otra aplicación la invoque en Web Apps 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.

Snapshot Debugger para .NET

La depuración de aplicaciones puede resultar difícil, especialmente si la aplicación se ejecuta en producción. Con Snapshot Debugger puedes tomar una instantánea de tus aplicaciones en producción cuando se ejecuta el código que te interese.

Este depurador te permite ver exactamente qué ha salido mal sin que esto afecte al tráfico de tu aplicación en producción. Snapshot Debugger puede ayudarte a reducir drásticamente el tiempo necesario para resolver los problemas que se producen en entornos de producción. Además, puedes usar Visual Studio para establecer puntos de anclaje y depurar paso a paso.

Parcheado automático del sistema operativo y .NET Framework

Dado que utilizas una plataforma totalmente administrada, no estarás gestionando tu propia infraestructura en absoluto y te aprovecharás del parcheado automático del sistema operativo (SO) y de .NET framework.

Máquinas virtuales

El alojamiento de tu aplicación en una VM en [Azure Virtual Machines](#) te ofrece mucho control sobre la forma de hospedar la aplicación. Sin embargo, serás el responsable del mantenimiento del entorno, incluido el parcheado del SO y la actualización de los programas antivirus.

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

Azure Functions

Con Azure Functions, puedes 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 Storage, Azure Cosmos DB y Azure Service Bus, e incluso servicios externos como Twilio y SendGrid. Con Functions, puedes crear funciones con rapidez y hospedarlas en un entorno versátil que realiza el escalado de forma automática.

Con Azure Functions es posible pagar únicamente por las funciones que se ejecuten en lugar de tener que mantener las instancias de computación en ejecución todo el mes. A esto se le llama también “sin servidor”, porque solo tienes que crear tu aplicación sin preocuparte por los servidores ni por el escalado de los mismos.

Puedes escribir Azure Functions en .NET, JavaScript, Java y una lista de lenguajes cada vez mayor.

Una aplicación que usa Functions activa una función cada vez que se carga un nuevo archivo de imagen en Azure Blob Storage. A continuación, la función cambia el tamaño de la imagen y escribe en otra cuenta de almacenamiento de Blob.

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

```
// Pruébalo: Crea tu primera Azure  
Function usando el portal de  
Azure.
```

Azure Logic Apps

Puedes organizar la lógica de negocio con [Logic Apps](#) mediante la automatización de un proceso de negocio o la integración 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. Al enlazar llamadas de la API a conectores, puedes 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](#) que pueden conectarse con Azure SQL Database, 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 Azure Functions, Logic Apps es una característica "sin servidor", se escala automáticamente y solo se paga por ella cuando se ejecuta.

A continuación hay un ejemplo que muestra un flujo de trabajo de Logic Apps:

1. La aplicación lógica se activa cuando llega un correo electrónico que contiene un pedido de envío a Office 365.
2. Con 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 mediante Twilio 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 Batch

Si necesitas ejecutar aplicaciones por lotes a gran escala o de computación de alto rendimiento (HPC) en máquinas virtuales, puedes usar [Azure Batch](#). Batch crea y administra una colección de miles de máquinas virtuales, instala las aplicaciones que quieras ejecutar y programa trabajos en dichas máquinas. No necesitas implementar ni gestionar máquinas virtuales individuales 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 hacen 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 on-premises, 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 similares que las máquinas virtuales, pero mucho más ligeros, y puedes iniciarlos y detenerlos en solo unos segundos.

Los contenedores también ofrecen una inmensa portabilidad, lo cual los hace ideales para desarrollar una aplicación de forma local y luego hospedarla en el cloud, en un entorno de prueba y después de 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.

Escala y coordina contenedores con Azure Kubernetes Service

[Azure Kubernetes Service](#) (AKS) simplifica crear, configurar y administrar un clúster de máquinas virtuales preconfiguradas para ejecutar contenedores. Eso significa que puedes usar tus habilidades para administrar e implementar aplicaciones que se ejecuten en contenedores en Azure.

AKS reduce la complejidad y la sobrecarga operativa de administrar un clúster de Kubernetes, ya que delega gran parte de esa responsabilidad en Azure. Al ser un servicio hospedado de Kubernetes, Azure administra tareas críticas, tales como la supervisión de estado y el mantenimiento. Además, solo pagas por los nodos de agente dentro de los clústeres, no por los maestros. Al ser un servicio gestionado de Kubernetes, AKS proporciona actualizaciones

de versiones y parches de Kubernetes de forma automática, fácil escalamiento de clústeres, plano de control hospedado de autorreparación (maestros), y ahorro de costes, ya que solo pagas por los nodos que se ejecutan del conjunto de agentes.

Dado que Azure se hace cargo de la administración de los nodos en tu clúster de AKS, hay muchas tareas que no tienes que realizar manualmente, como las actualizaciones de clústeres. Y puesto que Azure se hace cargo de estas importantes tareas de mantenimiento, AKS no ofrece acceso directo (como en SSH) al clúster.

// Más información: [Aprende a usar Azure Kubernetes Service](#)

Alojamiento de contenedores con Azure Container Instances

Puedes hospedar tu contenedor usando [Azure Container Instances](#) (ACI). ACI proporciona capacidad de cómputo aislado y rápido para atender el aumento repentino de tráfico sin la necesidad de administrar servidores. Por ejemplo, Azure Container Service (ACS) puede usar Virtual Kubelet para aprovisionar en ACI pods que pueden iniciarse en cuestión de segundos. Esto permite que ACS se ejecute con suficiente capacidad para una carga de trabajo promedio. Al agotarse la capacidad en tu clúster de ACS, puedes escalar con pods adicionales en ACI sin administrar ningún servidor adicional. El servicio ACI se factura por segundo, por CPU virtual, por gigabyte o por memoria.

// Más información: [Aprende más sobre Azure Container Instances](#)

Alojamiento de contenedores en Azure App Service Web App for Containers

[Web App for Containers](#) te ayuda 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. Puedes ejecutar contenedores de Docker (Linux) y contenedores de Windows en Web App for Containers.

Azure Container Registry

Tras haber creado una imagen de contenedor en la que ejecutar tu aplicación, puedes almacenar el contenedor en [Azure Container Registry](#) (ACR). Este es un servicio de almacenamiento seguro y de alta disponibilidad, creado específicamente para almacenar imágenes de contenedores. Es excelente para almacenar tus imágenes de Docker privadas.

También puedes usar ACR para tus canalizaciones de desarrollo e implementación de contenedores existentes. Puedes usar ACR Build para crear imágenes de contenedores en Azure.

Puedes crear versiones bajo demanda o automatizar totalmente las versiones con los disparadores de compilación de confirmación del código fuente y de actualización de la imagen de base.

Azure Service Fabric

Otra forma de ejecutar tus aplicaciones en Azure es ejecutarlas en [Azure Service Fabric](#). Este es el servicio que ejecuta muchas de los servicios de Azure dentro de Microsoft, como Azure SQL Database y Azure App Service. Ejecuta tus aplicaciones en Azure Service Fabric para lograr alta disponibilidad, a gran escala y realizar actualizaciones continuas.

Puedes utilizar Azure Service Fabric para ejecutar aplicaciones basadas en microservicios de .NET, que son soluciones que consisten en muchos servicios pequeños que hablan entre sí y son empleados por las interfaces de usuario y otros componentes. Service Fabric es ideal para este tipo de soluciones porque coordina los componentes de aplicaciones y los ejecuta con gran rendimiento y muy alta disponibilidad.

Azure Service Fabric es único, ya que puedes ejecutarlo en cualquier lugar. Instala Service Fabric en tu ordenador de desarrollo local, on-premises, o en cualquier cloud, incluido Azure. También puedes utilizar Azure Service Fabric Mesh para ejecutar contenedores en un clúster de Service Fabric que Microsoft administre como un servicio. Esto abre muchas posibilidades.

Es fácil implementar aplicaciones en Azure Service Fabric y administrarlas con tus herramientas favoritas, como Visual Studio y Azure DevOps Services. Además, Service Fabric ha pasado a ser código abierto hace poco.

Qué utilizar y cuándo

Algunos de los servicios que ejecutan tu aplicación en Azure pueden funcionar bien conjuntamente en una solución, mientras que otros son más adecuados para diferentes propósitos.

Aunque puede que esto complique la elección de los servicios adecuados, en la Tabla 1-1 te ayudaremos a identificar qué servicios de Azure son los adecuados para tu situación.

Tabla 1-1

	Web Apps*	Web App for Containers*	Mobile Apps*	Functions*	Logic Apps*	Virtual Machines*	Kubernetes Service*	Service Fabric*	Container Instances*	Batch*
Aplicaciones monolíticas y N-Tier		●				●**			●	●
Back end de app móvil			●			●**				
App de arquitectura de microservicios				●			●	●		
Orquestación y flujos de trabajo de procesos de negocio				●	●					
Tareas de computación intensiva										●
Ejecutar la aplicación en cualquier lugar (incluso on-premises)		●					●	●	●	

* Los servicios con un asterisco tienen un nivel gratuito que puedes utilizar para empezar gratis.

** Para migrar mediante lift-and-shift aplicaciones existentes a Azure.

Crear aplicaciones con mejor rendimiento

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 Traffic Manager

Muchas aplicaciones modernas tienen usuarios en todo el mundo. Proporcionar una experiencia de alto rendimiento para todos ellos es, como poco, un reto. El problema más obvio al que hay que enfrentarse es la latencia, es decir, el tiempo que tarda una señal o solicitud en llegar a 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 las aplicaciones de Web Apps. Web Apps actúa como punto de conexión, cuyo estado y rendimiento supervisa Azure Traffic Manager.

Cuando los usuarios acceden a tu aplicación, Traffic Manager los dirige a la aplicación de Web Apps que rinda mejor en su entorno cercano.

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

Azure Front Door

Puede que tus usuarios estén repartidos por el mundo, o que estén de viaje a veces. Esto puede hacer que sea difícil asegurarse de que experimenten un buen rendimiento y que tu aplicación sea segura y esté disponible independientemente de la ubicación.

[Azure Front Door](#) puede ayudar con esto. Este servicio puede dirigir el tráfico de los usuarios al punto de conexión con la aplicación que tenga mejor rendimiento, para que este mejore. Azure Front Door puede dirigir a puntos de conexión disponibles y evitar aquellos no disponibles.

Azure Traffic Manager también cumple esta función, pero de forma diferente a Azure Front Door. Azure Front Door funciona en la [capa 7 de OSI](#) o la capa HTTP/HTTPS, mientras que Azure Traffic Manager funciona con DNS. En otras palabras, Azure Front Door funciona en el nivel de la aplicación y Azure Traffic Manager en el nivel de red. Esta es una diferencia fundamental que determina las capacidades de los servicios.

Dada esta diferencia, Azure Front Door va mucho más allá de dirigir a los usuarios a puntos de conexión disponibles y con buen rendimiento.

Azure Front Door te permite modificar las reglas del firewall de aplicaciones web (WAF) para controlar el acceso y proteger tu carga de trabajo HTTP/HTTPS de los ataques basados en las direcciones IP clientes, el código del país y los parámetros HTTP.

Además, Front Door te permite crear reglas de limitación de la velocidad para combatir el tráfico de bots malintencionados. Estas son solo algunas de las capacidades exclusivas de Azure Front Door.

Entre el resto de capacidades de Front Door se incluyen:

- **Enrutamiento basado en URL**
Esto te permite dirigir las solicitudes de diferentes URL a distintos grupos del back end (aplicaciones que reciben tráfico, como Web Apps). Por ejemplo, `http://www.contoso.com/users/*` va a un grupo y `http://www.contoso.com/products/*` va a otro.
- **Reescritura de URL**
Esto te permite personalizar la URL que pasas al grupo del back end.
- **Terminación SSL**
Con esto, puedes proteger el tráfico de extremo a extremo, desde el navegador hasta la aplicación en el grupo del back end.
- **Afinidad de sesiones**
Cuando quieres que se envíe a los usuarios al mismo punto de conexión todas las veces, la afinidad de sesiones resulta muy útil. Esto es importante en aquellos casos en los que el estado de la sesión se guarda en el entorno local, en el back end, para la sesión de un usuario.

Si necesitas ayuda para elegir entre Azure Front Door y Traffic Manager, ten en cuenta esta tabla:

	Azure Traffic Manager	Azure Front Door
Solo necesitas enrutamiento (basado en el rendimiento o la ubicación) y alta disponibilidad	●	
Necesitas terminación SSL (también denominada descarga de SSL)		●
Necesitas funciones de la capa de aplicaciones, como la reescritura URL y el WAF		●

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 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, donde la referencia era `~/images/image.png`, ahora sería `https://example.azureedge.com/image.png`.

Esto no solo es muy fácil de hacer, sino que también mejora el rendimiento de la aplicación de las siguientes formas:

- Reduce el contenido suministrado desde la aplicación. Ya que ahora lo suministra Content Delivery Network, 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. Una forma de usar Content Delivery Network es servir vídeos para una aplicación móvil. Ya que los vídeos pueden ser grandes, no conviene almacenarlos en el dispositivo móvil (los usuarios no quieren hacerlo). Con Content Delivery Network, los vídeos se sirven desde el punto de presencia (PoP). Como está más cerca del usuario, mejora también el rendimiento.

// 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 en la caché.
3. La próxima vez que la aplicación busque esos datos, los 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 mejores SLA del sector. Es muy eficaz e incluye opciones avanzadas como la agrupación en clústeres y la replicación geográfica.

```
// Pruébalo: Empieza a trabajar  
con Azure Redis Cache
```

Lectura adicional

Si quieres obtener más información sobre el uso de Azure Kubernetes Service, Azure Container Instances y otros servicios de Azure para crear aplicaciones distribuidas, descarga y lee los siguientes e-books gratuitos:

```
// Incluir tus aplicaciones en  
contenedores con Docker y  
Kubernetes
```

```
// Diseño de sistemas distribuidos
```

02 /

Conectar tu aplicación con datos

¿Qué puede hacer Azure por tus datos?

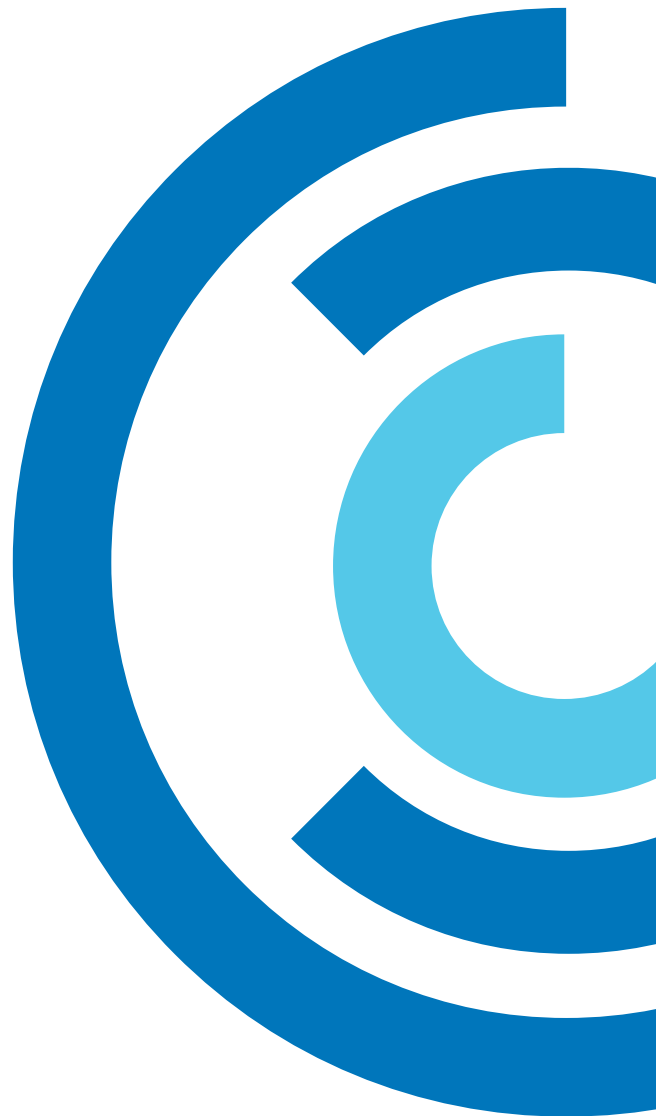
Allí donde estén tus datos, Azure te ayuda a aprovechar todo su potencial. Impulsa el crecimiento rápido y ahorra más tiempo para dedicarlo a la innovación con una cartera de servicios de base de datos seguros y de nivel empresarial compatibles con los motores de base de datos de código abierto.

Los servicios de base de datos de Azure están totalmente administrados, liberan tiempo muy valioso para que te puedas centrar en nuevas formas de servir a tus usuarios y en descubrir nuevas oportunidades, en lugar de dedicar ese tiempo a gestionar tu base de datos. El rendimiento de nivel empresarial con alta disponibilidad integrada supone que puedes escalar rápidamente y conseguir una distribución global sin preocuparte del costoso tiempo de inactividad.

Los desarrolladores pueden aprovecharse de innovaciones punteras en el sector, como la seguridad integrada, con supervisión automática y detección de amenazas, ajustes automáticos para un mejor rendimiento y distribución global inmediata. Además de todo esto, tu inversión estará protegida por SLA con respaldo financiero.

Creas lo que crees, te ayudaremos a llevarlo al mercado rápidamente, a distribuirlo ampliamente y a gestionarlo fácilmente y con confianza.

Pues en marcha.



Dónde almacenar los datos

Azure proporciona muchos tipos de almacenes de datos que pueden ayudarte a mantener y recuperar los datos en cualquier escenario. En la tabla 2-1 se presentan las opciones de almacenamiento disponibles en Azure.

Todos los servicios tienen un nivel gratuito que puedes utilizar para empezar.

// 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.

Tabla 2-1

	Base de datos SQL*	MySQL*	PostgreSQL*	MariaDB*	Azure Cosmos DB*	Blob*	Tabla*	Cola*	Archivo*	Disco*	Data Lake Store*	SQL Data Warehouse*
Datos relacionales	●	●	●	●							●	●
Datos no estructurados					●	●					●	
Datos semiestructurados					●		●					
Mensajes en cola								●				
Archivos en disco									●			
Archivos de alto rendimiento en disco										●		
Almacenamiento de datos grandes					●	●			●	●	●	●
Almacenamiento de datos pequeños	●	●	●	●	●	●	●	●	●	●		
Replicación de datos geográficos	●	●	●	●	●	●	●	●	●	●		
Consistencia de los datos ajustables					●							

* Los servicios con un asterisco tienen un nivel gratuito que puedes utilizar para empezar gratis.

Azure Cosmos DB

Azure Cosmos DB es un nuevo tipo de base de datos pensado para el cloud. Entre sus características destacables se incluyen:

- Un SLA del 99,99 % (99,999 % para operaciones de lectura) 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.
- [Niveles de consistencia de datos ajustables](#), para poder habilitar un verdadero sistema de datos distribuidos globalmente. Puedes elegir entre un amplio espectro de modelos de coherencia de datos, incluido el de consistencia fuerte, el de coherencia de sesión y el de coherencia de eventos.
- Traffic Manager, que envía a los usuarios al punto de conexión de servicio más próximo a ellos.
- Escala global sin límites, por lo que solo pagas por el rendimiento y el almacenamiento que necesitas.
- [Indexado automático de los datos](#), que elimina la necesidad de mantener o ajustar la base de datos.

Además de todas estas características, Azure Cosmos DB ofrece diferentes API con las que puedes almacenar y recuperar datos, como SQL, JavaScript, Gremlin, MongoDB, Almacenamiento de tablas de Azure y Apache Cassandra. Las distintas API gestionan los datos de diferentes formas. Puedes utilizar documentos como datos y también tablas no estructuradas, gráficos y blobs. Tú utilizas la API que se adapte a tus necesidades y Azure Cosmos DB se encarga del resto.

Disfrutarás de rendimiento de nivel de cloud, escalabilidad y fiabilidad mientras sigues usando el modelo de programación al que ya estás acostumbrado.

```
// Pruébalo: Introducción a Azure  
Cosmos DB
```

Azure SQL Database

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

Con SQL Database, puedes hacer [casi lo mismo](#) que con SQL Server on-premises. De hecho, las nuevas características de SQL Server primero se integran en Azure SQL Database y después en SQL Server on-premises.

Puedes utilizar SQL Database con tus herramientas favoritas, como SQL Server Management Studio y Entity Framework. Las bases de datos de SQL Database son extremadamente fiables y sólidas y ofrecen un SLA que garantiza un tiempo de actividad del 99,99 %.

A continuación se describen las características más avanzadas de SQL Database:

- [Replicación geográfica](#), que replica los datos en otras regiones geográficas en tiempo real.
- [Enmascaramiento dinámico de datos](#), que enmascara dinámicamente los datos confidenciales para determinados usuarios en tiempo de ejecución.
- [Auditoría](#), que proporciona una auditoría completa de todas las acciones que se realizan en los datos.

- [Ajuste automático de base de datos](#), que supervisa el rendimiento de tu base de datos y la ajusta de forma automática.

SQL Database ofrece varios niveles de servicio orientados para diferentes escenarios.

- **Fines generales/estándar:** este nivel ofrece opciones de computación y almacenamiento equilibradas y escalables para presupuestos restringidos. Este nivel está totalmente administrado y ofrece un rendimiento comparable al de las VM de Azure SQL, por lo que es la mejor opción para la mayoría de cargas de trabajo empresariales.
- **Esencial para el negocio/premium:** este nivel ofrece la máxima resiliencia ante errores con varias réplicas aisladas. Con una E/S constantemente alta, incluye un grupo de disponibilidad para alta disponibilidad integrado. Esta es la mejor opción para las aplicaciones empresariales de procesamiento de transacciones online (OLTP) críticas (operaciones CRUD normales) con requisitos de E/S constantemente altos.
- **Hiperescala:** este nivel ofrece compatibilidad con bases de datos de gran tamaño (VLDB) sin complicaciones. Con una arquitectura creada para el cloud de almacenamiento de alta escalabilidad y un caché multicapa optimizado para cargas de trabajo muy voluminosas y exigentes, ofrece una baja latencia y un rendimiento alto independientemente del tamaño de las operaciones de datos. Este es el mejor nivel para cargas de trabajo muy grandes y exigentes con requisitos de almacenamiento y lectura-escala altamente escalables.

Bases de datos de Azure para MySQL, PostgreSQL y MariaDB

Azure proporciona bases de datos [MySQL](#), [PostgreSQL](#) y [MariaDB](#) como bases de datos administradas, lo cual significa que solo las activa y no tiene que preocuparse por la infraestructura subyacente. Al igual que Azure SQL Database y Azure Cosmos DB, estas bases de datos están disponibles universalmente, son escalables, muy seguras y completamente administradas.

Cada una de estas bases de datos es adecuada para casos de uso un poco distintos, pero en términos generales sus funcionalidades se complementan mucho. Usarías las bases de datos de Azure for MySQL, PostgreSQL y MariaDB cuando ya uses una de sus versiones on-premises y busques aprovechar la administración completa en el cloud.

Azure Storage

[Azure Storage](#) 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 cinco tipos de Azure Storage son Blob, tabla, cola, archivo y disco (Figura 2-1).



Figura 2-1

Alojamiento de sitios web estáticos en Azure Storage

Otra función emocionante de Azure Storage es [el alojamiento de sitios web estáticos](#). Esta función de sitios web estáticos solo utiliza Blob Storage como almacén de datos, y puedes utilizarlo para hospedar un sitio web estático en Azure Storage. Todo lo que tienes que hacer para que tu sitio web se ejecute es cargar los archivos de tu sitio web estático en Blob Storage e indicar qué archivo es el documento predeterminado (como index.html) y cuál es el de error (como 404.html). Tu sitio web se ejecutará rápido a un coste muy reducido: de hecho, solo pagarás por el almacenamiento que uses, y la función de sitio web estático no tiene ningún coste adicional. Además, cuando utilizas la redundancia geográfica (habilitada de forma predeterminada), tu sitio web estará listo incluso aunque tu centro de datos principal falle.

Blob Storage

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

Hay tres tipos de blobs: [blobs de página](#), [blobs de anexo](#) y [blobs en bloques](#). Los blobs de página están optimizados para las operaciones aleatorias de lectura y escritura y son perfectos para almacenar un VHD. 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. Los blobs de anexo están optimizados para las operaciones de anexo, como el almacenamiento de registros de operaciones que no se puedan actualizar o eliminar.

```
// Pruébalo: Empieza a usar  
Azure Blob Storage
```

Table Storage

[Azure Table Storage](#) es un servicio de almacenamiento de valores clave NoSQL extremadamente rápido y barato que puedes usar para almacenar datos en tablas flexibles. Una tabla puede contener una fila que describe un pedido y otra fila que describe la información del cliente. No necesitarás definir un esquema de datos, por lo que el almacenamiento en tablas es muy flexible.

```
// Pruébalo: Empieza a usar  
Azure Table Storage
```

Queue Storage

[Azure Queue](#) Storage es un tipo de almacenamiento inusual. Aunque se utiliza para almacenar pequeños mensajes de datos, su principal objetivo es servir como cola. Los mensajes se ponen en la cola y otros procesos los recogen. [Este patrón](#) desvincula el remitente del procesador de mensajes, por lo que mejora el rendimiento y la fiabilidad. Azure Queue Storage se encuentra 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), por lo que se puede utilizar con Windows y Linux, y se puede tener acceso a él desde el cloud o desde sistemas on-premises. Al igual que otros servicios en 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 para ejecutar SQL Server, por ejemplo. Disk Storage 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](#), por otra parte, es un almacenamiento para aplicaciones de Big Data. Se puede usar para almacenar grandes cantidades de datos en su formato nativo: estructurados, no estructurados y de 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 Store:

- 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.

Puedes almacenar todos los datos que obtienes de los dispositivos del Internet de las cosas (IoT) que están recopilando datos de temperatura, por ejemplo, en Data Lake Storage. 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 Storage 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 el portal  
de Azure
```

Azure SQL Data Warehouse

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

Dado que SQL Data Warehouse se ejecuta en Azure, tienes muchas funciones disponibles. Una de ellas es 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.

Una forma eficaz de usar SQL Data Warehouse es conocer 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
```

Soluciones de análisis de datos de Azure

Casi tan importante como almacenar datos es analizarlos para obtener conocimientos. Azure proporciona muchos servicios para escenarios de análisis de datos, lo que le permite obtener conocimientos valiosos y útiles de sus datos, no importa qué tan grandes, pequeños o complejos sean.

Azure Data Factory

Trasladar y transformar los datos no es una tarea trivial, pero [Azure Data Factory](#) puede ayudarte a hacerlo. Dentro de Data Factory puedes crear un proceso integral que realice todo el proceso de extracción, transformación y carga (ETL).

Data Factory puede trasladar los datos de forma fiable de on-premises al cloud, dentro del cloud o a on-premises. No importa dónde se encuentren las fuentes de tus datos. Data Factory también ofrece muchos conectores que puedes utilizar para conectarse fácilmente con tus fuentes de datos, tales como SQL Server, Azure Cosmos DB, Oracle y [muchos más](#).

Cuando mueves datos, también puedes filtrarlos antes de enviarlos a un destino final, limpiarlos o transformarlos con una actividad en el proceso, tal como [Apache Spark Activity](#). Además, Azure Data Factory te permite programar y supervisar las canalizaciones, además de trasladar mediante [lift and shift tus paquetes de servicios de integración de SQL Server \(SSIS\)](#) al cloud.

// Pruébalo: Crear una fábrica de datos con el portal de Azure

Azure Analysis Services

Con [Azure Analysis Services](#), puedes crear un modelo semántico de tus datos al que los usuarios pueden obtener acceso directamente con herramientas de visualización como Power BI. El servicio, que está creado sobre las herramientas de [SQL Server Analysis Services](#) que se ejecutan on-premises con SQL Server, ahora se ejecuta administrado en el cloud. Esto implica que el servicio es escalable y que los datos se almacenan de forma redundante: cuando no los usas, puedes pausar el servicio para minimizar costes.

Con Azure Analysis Services, puedes proporcionar datos modelados directamente a los usuarios de forma muy eficiente. Los usuarios pueden consultar millones de registros en segundos, ya que el modelo vive completamente in-memory y se actualiza periódicamente.

Puedes obtener datos en el modelo semántico desde cualquier lugar, incluso desde fuentes de datos en el cloud y on-premises. Puedes utilizar Azure Blob Storage, Azure SQL Databases, Azure SQL Data

Warehouse y [muchos otros servicios](#) como fuentes de datos para el modelo. También puedes utilizar fuentes de datos como las bases de datos on-premises de Active Directory, Access y Oracle.

```
// Pruébalo: Crear un servidor  
Azure Analysis Services con  
el portal de Azure
```

Azure Data Lake Analytics

Otro servicio de Azure para realizar tareas de análisis de datos es [Azure Data Lake Analytics](#). Con este servicio, puedes analizar, procesar y transformar cantidades potencialmente masivas de datos de Azure Storage y Azure Data Lake Store.

Azure Data Lake Analytics te permite crear y enviar tareas que consultan datos, los analizan o los transforman. Puedes escribir estas tareas en U-SQL, que es un lenguaje tipo SQL, y expandir U-SQL con Microsoft R y Python.

Pagas por las tareas que envías y ejecutas, y el servicio se escala de forma automática dependiendo de la capacidad que requieran las tareas. Por lo general, Azure Data Lake Analytics se utiliza para tareas de análisis de largo plazo en grandes cantidades de datos.

```
// Pruébalo: Crea tu primer script  
U-SQL con el portal de Azure
```

Azure Stream Analytics

Puedes utilizar el servicio [Azure Stream Analytics](#) para analizar, consultar y filtrar datos de streaming en tiempo real. Por ejemplo, cuando recibes un stream de datos de temperatura desde un dispositivo de IoT, te dice qué tan cálido está afuera. Podría proporcionar la misma temperatura cada segundo durante una hora hasta que la temperatura cambie, pero solo estás interesado en los cambios. Azure Stream Analytics puede consultar los datos en tiempo real y almacenar solo los datos diferenciales en Azure SQL Database.

Stream Analytics puede obtener datos desde muchos servicios, incluido Azure Blob Storage, Azure Event Hubs o Azure IoT Hub. Puedes analizar los datos utilizando un lenguaje simple estilo SQL o un código personalizado. Después de consultar y filtrar el stream de datos, Stream Analytics puede producir el resultado en muchos servicios de Azure, incluidos Azure SQL Database, Azure Storage y Azure Event Hubs.

```
// Pruébalo: Crea una tarea de  
Stream Analytics con el portal  
de Azure
```

Azure Time Series Insights

Puedes utilizar [Azure Time Series Insights](#) para obtener información rápida sobre grandes cantidades de datos, típicas del IoT. Este servicio obtiene datos desde Azure Event Hubs, IoT Hub y tus propias entradas de referencia, y conserva dichos datos durante un tiempo determinado.

Con Azure Time Series Insights, los usuarios pueden consultar y analizar datos mediante la herramienta de visualización en cuanto llegan. Time Series Insights no solo analiza los datos, también los introduce y conserva por un tiempo. Es similar a Azure Analysis Services, donde los datos residen in-memory en un modelo para que los usuarios puedan consultarlos. Las principales diferencias son que Time Series Insights está optimizado para IoT y datos basados en tiempo y contiene su propia herramienta de visualización de datos.

```
// Pruébalo: Explora una demostración del entorno Time Series Insights desde tu navegador
```

Azure Databricks

[Azure Databricks](#) te permite ejecutar en el cloud un clúster de Databricks administrado y escalable. Databricks proporciona una plataforma unificada de análisis con una gran variedad de herramientas y capacidades. En Databricks, puedes ejecutar versiones optimizadas de Apache Spark para hacer análisis avanzado de datos.

Además del análisis basado en Spark, Databricks ofrece notas interactivas y flujos y espacios de trabajo integrados que puedes utilizar para colaborar con todo el equipo de datos, incluidos los científicos de datos, ingenieros de datos y analistas de negocios, los cuales tienen acceso a herramientas especializadas para sus necesidades específicas.

Databricks está completamente integrado con Azure Active Directory, lo cual permite implementar una seguridad granular. Con Databricks, puedes realizar un análisis de datos basados en Spark con datos que provienen de muchos lugares, tales como

Azure Storage y Azure Data Lake Store. Databricks también funciona con datos de Azure SQL Data Warehouse, Azure SQL Database y Azure Cosmos DB. Además, puedes conectar Databricks a Power BI para crear y mostrar paneles de gran calidad.

```
// Pruébalo: Ejecuta una tarea de Spark en Azure Databricks con el portal de Azure
```

Azure HDInsight

[Azure HDInsight](#) es una plataforma dentro de Azure que puedes usar para ejecutar servicios de código abierto para análisis de datos. Puedes utilizarlo también para ejecutar clústeres especializados de tus herramientas favoritas de código abierto para análisis de datos. La ventaja de ejecutar estas herramientas en Azure HDInsight es que son administradas, lo que significa que no tienes que mantener máquinas virtuales ni actualizar sistemas operativos. Además, pueden escalarse y conectarse con facilidad entre sí, con otros servicios de Azure y con los servicios y fuentes de datos on-premises.

La mayoría de los tipos de clústeres de análisis de datos de código abierto especializados en Azure HDInsight utilizan Azure Blob Storage o Azure Data Lake Store para obtener acceso a los datos o almacenarlos, dado que estos servicios funcionan con el sistema de archivos Hadoop.

Potencialmente, puedes ejecutar distintos tipos de clústeres masivos especializados, como un **clúster Apache Hadoop**. Esto te permite procesar y analizar datos con herramientas de Hadoop como Hive, Pig y Oozie.

También puedes lanzar un **clúster Apache HBase**, el cual proporciona una base de datos NoSQL muy veloz. Los datos se encuentran en realidad dentro de Azure Storage o Azure Data Lake, pero HBase proporciona una capa de abstracción encima, la cual tiene su propia funcionalidad y rendimiento único.

Puedes crear un **clúster Apache Storm**, que está diseñado para analizar streams de datos, como Azure Stream Analytics. Además, puedes tener un **clúster Apache Spark**, que ofrece un marco para procesar y analizar enormes cantidades de datos. HDInsight también puede ejecutar un clúster para **Microsoft Machine Learning Server** (anteriormente, servidor Microsoft R).

Esto te permite ejecutar tareas basadas en R para analizar datos. Por último, puedes crear un clúster que ejecute **Apache Kafka**, el cual es un sistema de mensajes de suscripción-publicación que se utiliza para desarrollar aplicaciones con mecanismos de cola.

Hay más tipos de clúster, así como herramientas que puedes usar dentro de los clústeres. Puedes realizar casi cualquier tarea de análisis y procesamiento de datos con una combinación de estos clústeres, y todos se administran en el cloud. La Tabla 2-2 puede ayudarte a escoger los servicios de Azure adecuados para analizar los datos.

// Pruébalo: Extrae, transforma y carga datos con Apache Hive en Azure HDInsight

Tabla 2-2

	Data Factory*	Analysis Services*	Data Lake Analytics*	Stream Analytics*	Time Series Insights*	Azure Databricks*	Azure HDInsight*
Mover datos de almacén a almacén	●						
Transformar datos	●	●	●	●	●	●	●
Consultar y filtrar datos de streaming				●		●	●
Proporcionar un modelo semántico in-memory para los usuarios		●			●		●
Permitir a los usuarios consultar datos y crear paneles					●		
Analizar datos para uso posterior			●		●		●

* Los servicios con un asterisco tienen un nivel gratuito que puedes utilizar para empezar gratis.

Tutorial: Publicación de un sitio web on-premises en Azure con una base de datos SQL

Antes de comenzar, necesitarás:

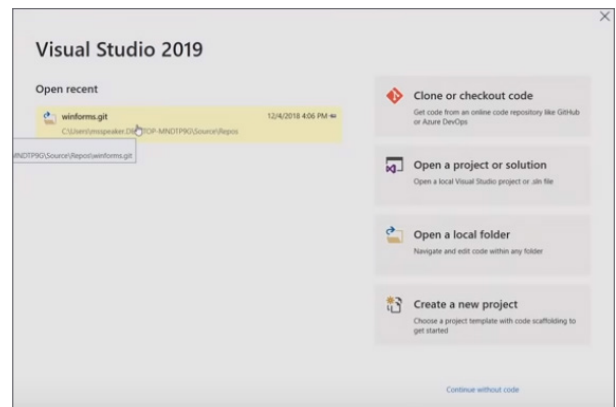
- Microsoft Visual Studio 2017 o posterior
- Una [cuanta gratuita de Azure](#) para poder seguir este script de demostración
- [Código fuente de Tailwind Traders Rewards](#)

Tutorial: Migración de una aplicación .NET a Azure sin cambiar el código

Cuando tu aplicación crece más que tu infraestructura local, tendrás que analizar el resto de opciones para afrontar las demandas sin sobrecargar a los equipos.

El cloud de Azure ofrece varias plataformas y servicios para hospedar aplicaciones. Para empezar, utilizarás **Azure App Service** para hospedar la aplicación sin realizar cambios en el código existente

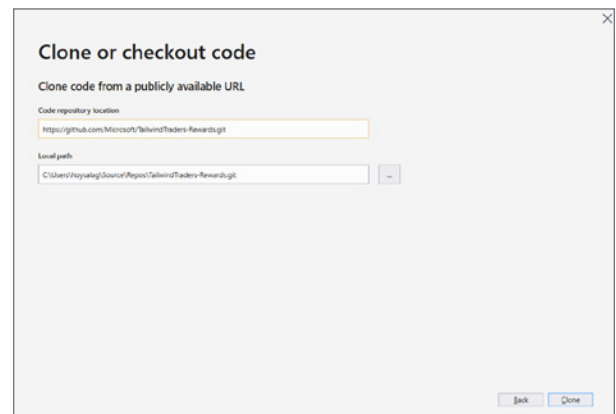
1. Lanza **Visual Studio 2019**. Verás inmediatamente la experiencia de “apertura” simplificada.



Lanzamiento de Visual Studio

2. Haz clic en la opción **Clonar o desproteger código** e introduce la URL del repositorio **Tailwind Traders Rewards** (<https://github.com/Microsoft/TailwindTraders-Rewards.git>) en la ubicación del repositorio del código.

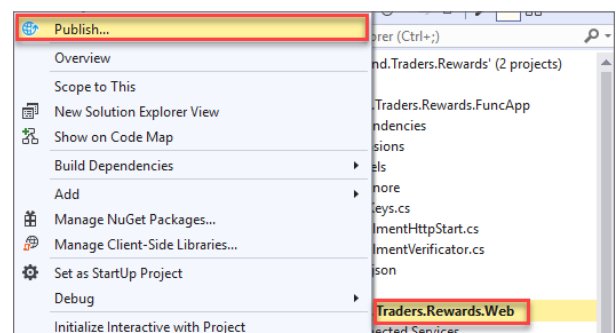
Haz clic en **Clonar**. En **Soluciones y carpetas**, haz clic en **Tailwind.Traders.Rewards.sln** para abrir la solución.



Clonación de Rewards

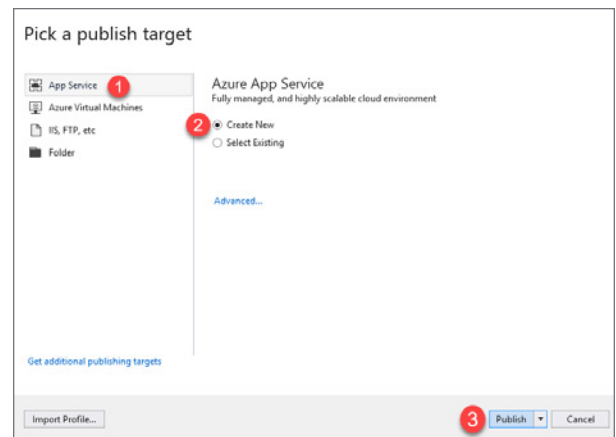
3. Haz clic derecho en el proyecto **Tailwind.Traders.Rewards.Web** y selecciona **Publicar**. Es el mismo cuadro de diálogo Publicar que puedes usar para implementar en **IIS6** en tu infraestructura local.

Mediante este cuadro de diálogo Publicar, podrás implementar la aplicación en la plataforma de Azure en el cloud.



Publicación de la aplicación

4. Elige **App Service** como objetivo de la Publicación. Bajo la ventana **Azure App Service**, selecciona **Crear nuevo** y haz clic en **Publicar**.

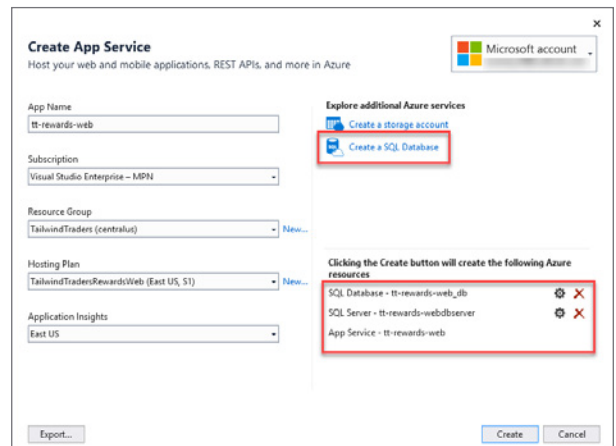


Opciones de publicación

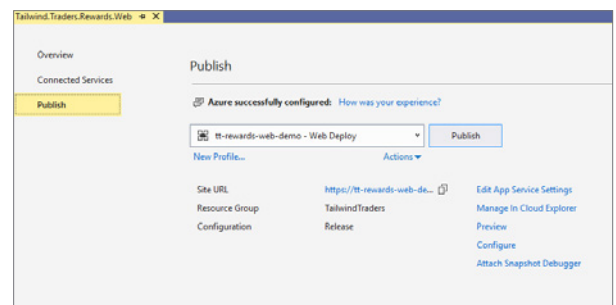
5. En la siguiente ventana, introduce la información de tu suscripción a Azure, selecciona **App Service** y elige las opciones existentes para **Grupo de recursos**, **Plan de hospedaje** y **Application Insights** o crea algunas nuevas.

Haz clic en la opción **Crear una base de datos SQL** en el lado derecho y crea un nuevo servidor y una nueva base de datos en las ventanas emergentes.

Por último, haz clic en **Crear** para **crear** un perfil de publicación. De forma alternativa, puedes crear la base de datos de Azure SQL directamente en el portal de Azure.



Creación del perfil

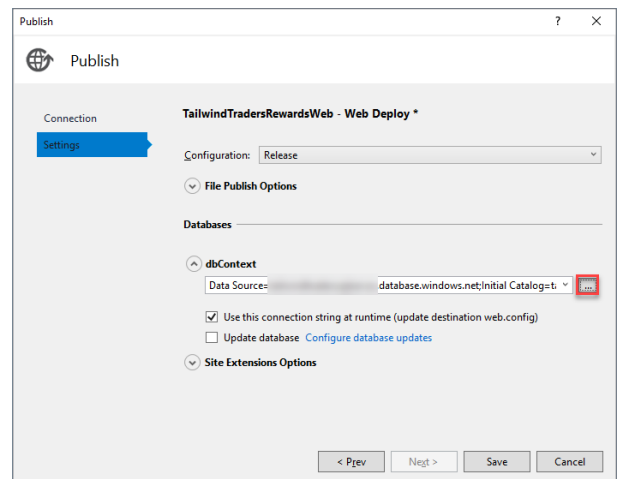


Creación del perfil

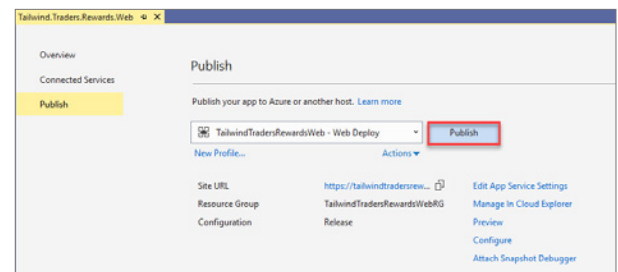
6. Haz clic en **Configurar** en la ventana Publicar para comprobar las cadenas de conexión de la base de datos. Las cadenas de conexión de la base de datos se pueden rellenar seleccionando el botón de los puntos suspensivos e introduciendo los detalles de la base de datos SQL. Al hacer clic en **Publicar**, el archivo de configuración web se actualizará con esta cadena de la base de datos, que dirige a una base de datos SQL.

Cuando se depura la aplicación de forma local, actuarán los servicios de información de Internet (IIS) y el SQL Server local, pero cuando se publica la aplicación, se intercambiará con los servicios de Azure creados.

7. Haz clic en **Publicar** para implementar la aplicación en Azure App Service y el back end en la base de datos SQL.

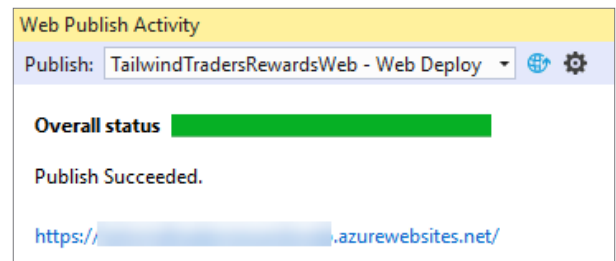


Cadenas de conexión a base de datos

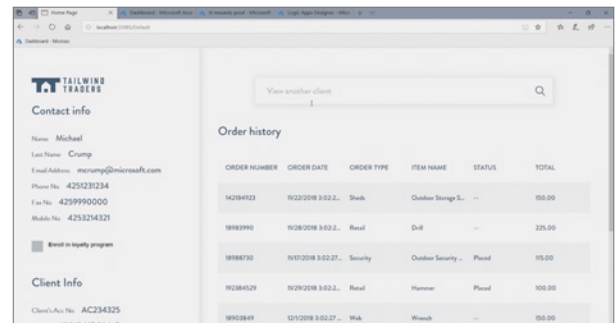


Publicación de la aplicación

8. Cuando se haya publicado la aplicación, verás su estado como **Publicación correcta**, y la aplicación web se abre en el navegador. Ahora el sitio web mostrará los datos de la base de datos SQL.



Publicación correcta



Aplicación web

Lectura adicional

Si quieres leer más acerca de los datos y el análisis de datos en Azure, puedes descargar y leer los siguientes e-books gratuitos:

// Guía de NoSQL con Azure Cosmos DB

// Azure para arquitectos

// Migración de aplicaciones .NET a Azure

03 /

Proteger una aplicación

¿Cómo puede Azure ayudar a proteger tu aplicación?

¿Alguna vez has tenido un incidente de seguridad con una de tus aplicaciones? Puede que hayas tenido alguno y no te hayas enterado. Con Azure, puedes proteger los datos, las aplicaciones y la infraestructura con servicios de seguridad integrados que incluyen inteligencia de seguridad para ayudarte a identificar amenazas en rápida evolución de forma anticipada, para que puedas responder rápidamente.

Azure también puede ayudarte a implementar una estrategia de defensa por capas y en profundidad en los ámbitos de identidad, datos, hospedaje y redes. Con servicios como el [Azure Security Center](#), podrás ver información general sobre tu estado de seguridad y obtener recomendaciones sobre cómo mejorar la seguridad.

Y lo más importante es que se te notificará en cuanto se detecte un incidente de seguridad, por lo que siempre estarás al tanto si hay alguna amenaza. De esta forma, puedes dar pasos inmediatos para proteger tus activos. En este capítulo, analizaremos algunos de ellos.

Azure Active Directory

Una parte importante de la seguridad de tu aplicación es la autenticación de los usuarios antes de utilizarla, pero la autenticación no es algo 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](#) (Azure AD) proporciona todo esto y más “out of the box”. Las identidades de los usuarios se guardan en Azure AD y los usuarios se autentican en él, de modo que solo se les redirige a la aplicación una vez que se han autenticado. Azure AD se encarga de administrar las contraseñas, lo que incluye la resolución de situaciones frecuentes como los olvidos de contraseñas.

Dado que millones de aplicaciones usan Azure AD cada día, entre ellas el [portal de Azure](#), [Outlook.com](#) y [Office 365](#), es capaz de detectar el comportamiento malintencionado y actuar de forma más ágil. Por ejemplo, si un usuario inicia sesión en una aplicación desde una ubicación en Europa y un minuto más tarde inicia sesión desde Australia, Azure AD marca esto como un comportamiento malintencionado y pide al usuario credenciales adicionales por medio de una autenticación multifactor.

Azure Key Vault

Dentro de tu arquitectura de seguridad, necesitas un lugar seguro para almacenar y administrar certificados, claves y otros secretos. [Azure Key Vault](#) proporciona esta capacidad. Con Key Vault, puedes almacenar los secretos utilizados por tus aplicaciones en una ubicación central.

```
// Pruébalo: Introducción a Azure  
Key Vault.
```

Entre estos secretos se pueden incluir las credenciales de una cadena de conexión. Tu aplicación obtendría la cadena de conexión 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 almacena SSL y otros certificados que se usan para proteger el tráfico intercambiado por las aplicaciones a través de HTTPS.

Azure Sentinel

Para obtener una buena información general del estado de seguridad de tu organización y todos sus usuarios, aplicaciones, servicios y datos, puedes usar una plataforma de información de seguridad y administración de eventos (SIEM). Ahora, Azure ofrece un SIEM basado en IA, [Azure Sentinel](#).

Puedes usar Azure Sentinel para recopilar datos de tu organización, incluidos datos sobre usuarios, aplicaciones, servidores y activos de infraestructura como cortafuegos y dispositivos que se ejecutan en el cloud y on-premises. Es sencillo recopilar datos de tu organización con los conectores integrados. A medida que los datos se recopilan, Azure Sentinel detecta las amenazas de seguridad y reduce al mínimo los falsos positivos con sus algoritmos inteligentes de machine learning.

Cuando existe una amenaza, se te alertará y podrás investigarlo con IA, recurriendo a las décadas de trabajo en ciberseguridad de Microsoft. Puedes responder a los incidentes con la orquestación de flujos de trabajo integrada y la automatización de tareas de Azure Sentinel.

```
// Empieza a incorporar Azure  
Sentinel
```

Azure API Management

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

Con API Management, puedes proteger una API pidiendo a los usuarios que se suscriban a ella. De esta forma, las aplicaciones 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.

Mucho más que seguridad

Aunque la seguridad es crucial, Azure API Management ofrece otras capacidades que pueden ayudar a agilizar tu flujo de desarrollo y de pruebas, tales como [simulacros de respuesta de datos de prueba](#), [publicación de varias versiones de API](#), [introducción de cambios seguros sin interrupciones con las revisiones](#) y otorgar acceso a la documentación autogenerada de API, catálogo de API y muestras de código para desarrolladores.

```
// Pruébalo: Introducción a Azure  
API Management.
```

Azure AD Application Proxy

[Azure AD Application Proxy](#) proporciona acceso de inicio de sesión único (SSO) y remoto seguro para apps web hospedadas on-premises. Las aplicaciones que seguramente querrías publicar incluyen sitios de SharePoint, Outlook Web Access y otras aplicaciones web de línea de negocios (LOB). Estas apps web on-premises se integran con Azure AD, la misma identidad y plataforma de control que utiliza Office 365. Los usuarios finales obtienen acceso a tus apps on-premises de la misma forma en que obtienen acceso a Office 365 y a otras aplicaciones SaaS integradas con Azure AD.

Identidades administradas para recursos de Azure

¿Cómo evitas que las credenciales sean localizables dentro del código? Utilizar Azure Key Vault es un comienzo, pero ¿dónde almacenas las credenciales para conectarte a Key Vault?

[Las identidades administradas para recursos de Azure](#) ofrecen una solución.

Puedes utilizar las identidades administradas para [muchos servicios de Azure](#), incluido Azure App Service. Simplemente habilitas las identidades administradas con un botón para inyectar las credenciales a tu aplicación en el momento de ejecución y después utilizas esas credenciales para obtener acceso a otros servicios, como Azure Key Vault. Todos los procesos de autenticación entre servicios se realizan en el nivel de infraestructura, por lo que tu aplicación no tendrá que encargarse de ellos y podrá usar otros servicios.

```
// Pruébalo: Cómo utilizar las  
identidades administradas para  
recursos de Azure en App Service  
y Azure Functions
```

Cifrado

Cifrado predeterminado de datos

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. Este cifrado se realiza automáticamente y no necesitas configurar nada para utilizarlo.

Para ayudar a satisfacer tus requerimientos de seguridad y cumplimiento, puedes utilizar las siguientes funciones para cifrar datos en reposo:

- [Azure Disk Encryption](#) cifra volúmenes de datos, arranque de máquinas virtuales e infraestructura como servicio (IaaS) de Windows y Linux utilizando claves administradas por los clientes.
- [Azure Storage Service Encryption](#) cifra los datos de forma automática antes de persistir en Azure Storage, y después descifra los datos automáticamente cuando los recuperas.
- [Azure Client-Side Encryption](#) es compatible con cifrado de datos dentro de las aplicaciones cliente antes de cargar a Azure Storage u otras terminales, y después descifra los datos al descargarlos al cliente.
- [SQL Transparent Data Encryption](#) (TDE) cifra los archivos de datos de [SQL Server](#), [Azure SQL Database](#) y [Azure SQL Data Warehouse](#). Los archivos de registro y los datos se cifran utilizando algoritmos de cifrado estándar de la industria. Las páginas en una base de datos se cifran antes de escribirse en el disco y se descifran al leerlas.
- SQL [Always Encrypted](#) cifra los datos dentro de aplicaciones cliente antes de almacenarlos en Azure SQL Database. Permite delegar la administración de bases de datos on-premises a terceros, y mantiene una separación entre quienes son propietarios y pueden ver los datos y quienes la administran pero no deben tener acceso a los mismos.
- [Azure Cosmos DB](#) no requiere ninguna acción de tu parte porque los datos del usuario almacenados en Azure Cosmos DB en un almacenamiento no volátil (unidades de estado sólido) están cifrados de forma predeterminada y no existen controles para modificar esa configuración.

Azure Security Center

[Azure Security Center](#) proporciona administración unificada de la seguridad y Advanced Threat Protection en todas las cargas de trabajo del cloud híbrido. Ofrece controles de política centralizados para limitar la exposición a amenazas y encontrar y solucionar las vulnerabilidades de forma rápida.

Además, Security Center es compatible con la integración de soluciones de terceros y puede personalizarse con capacidades de programación y automatización. Puedes usar Security Center para analizar el estado de la seguridad de tus recursos de computación, redes virtuales, servicios de almacenamiento y datos y aplicaciones.

Una evaluación continua ayuda a descubrir posibles problemas de seguridad, tales como sistemas con actualizaciones de seguridad faltantes o puertos de red expuestos. Una lista de recomendaciones y descubrimientos priorizados puede activar alertas y otros remedios guiados.

Protección de Azure contra ataques de denegación de servicio distribuidos

Lo has oído en las noticias y sin duda no deseas que le ocurra a tu empresa: una aplicación es blanco de 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 usarla. El [servicio de protección Azure DDoS](#) ofrece protección contra ataques DDoS en un nivel básico (el nivel Básico) y un nivel de pago (el nivel Estándar).

No tienes por qué hacer nada para activar el nivel Básico: se activa automáticamente para todos los clientes como parte de la plataforma Azure. Este servicio protege tus aplicaciones contra los ataques de DDoS más comunes mediante la realización de supervisión y mitigación en tiempo real, y proporciona las mismas defensas utilizadas por Microsoft Online Services (MOS).

El nivel Estándar proporciona capacidades adicionales de mitigación que se ajustan específicamente a los recursos de Azure Virtual Network. Es fácil de activar y no tienes que cambiar tus aplicaciones: todo se hace en el nivel de red. Además, con el nivel Estándar puedes personalizar la protección de nivel Básico con tus propias políticas que se centran en tus aplicaciones y casos prácticos específicos.

// Más información: Más información sobre la protección de Azure frente a DDoS

Azure VPN Gateway

Una de las diversas opciones para conectar Azure a tu red on-premises es [Azure VPN Gateway](#). Esto te permite configurar una conexión VPN cifrada Site-to-Site (S2S) entre una red virtual de Azure y tu red on-premises.

Dado que el tráfico está cifrado, es seguro, incluso si viaja por el Internet público. VPN Gateway puede enviar tráfico cifrado entre las redes virtuales de Azure por la red de Microsoft.

También puedes crear conexiones cifradas Point-to-Site (P2S) desde tu equipo a Azure. Así, tendrás tu propia conexión privada y segura a Azure incluso cuando estés en movimiento.

```
// Empieza creando una Azure VPN  
Gateway con PowerShell
```

Azure Application Gateway

[Azure Application Gateway](#) es un dispositivo virtual dedicado que proporciona un controlador de entrega de aplicaciones (ADC) como servicio. Ofrece varias capacidades de balance de carga Layer 7 para tu aplicación y permite que los clientes optimicen la productividad de la granja web al descargar terminación SSL de uso intensivo de CPU a la puerta de enlace de la aplicación. La puerta de enlace también proporciona otras capacidades de enrutamiento Layer 7, tales como distribución round-robin de tráfico entrante, afinidad de sesión basada en cookies, enrutamiento basado en ruta URL y la capacidad de hospedar varios sitios web tras una sola puerta de enlace de aplicación.

Firewall de aplicaciones web de Azure

Necesitas proteger tu aplicación frente a muchas amenazas, incluidas la inyección SQL y Cross-Site Scripting (XSS), además de otras definidas en Open Web Application Security Project (OWASP). [El firewall de aplicaciones web \(WAF\) de Azure](#) puede ayudarte con eso. El firewall de aplicaciones web es una característica del servicio [Azure Application Gateway](#) y proporciona protección en tiempo real para tu aplicación. Detecta ataques malintencionados, tal y como se define en el [conjunto de reglas básicas de OWASP](#), e impide que dichos ataques lleguen 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, dando una capa adicional de seguridad.

Azure Network Watcher

[Azure Network Watcher](#) es un servicio regional que te permite supervisar y diagnosticar condiciones en el nivel de la red, desde y hasta Azure.

Sus muchas herramientas de diagnóstico y visualización pueden ayudar a comprender y obtener conocimiento más detallado sobre la red en Azure.

Ejemplos:

- [Topología](#): Proporciona una visualización de la red que muestra las distintas interconexiones y asociaciones entre los recursos de red en un grupo de recursos.
- [Captura variable de paquete](#): Captura datos de paquetes dentro y fuera de una máquina virtual. Las opciones avanzadas de filtrado y controles de ajuste fino, como la capacidad de definir limitaciones de tamaño y hora, proporcionan versatilidad. Los datos del paquete pueden almacenarse en blob o en el disco local en formato .cap.
- [Verificación de flujo de IP](#): Confirma si un paquete se autoriza o no con base en los parámetros de paquete de tupla de 5 elementos (IP destino, IP origen, Puerto destino, Puerto origen y protocolo). Si un grupo de seguridad rechaza el paquete, se obtienen la regla y el grupo que rechazaron el paquete.

Grupos de seguridad de red

Un [grupo de seguridad de red](#) (NSG) contiene una lista de reglas de seguridad para permitir o rechazar el tráfico de red a los recursos conectados a Azure Virtual Networks. Los NSG pueden asociarse con subredes, máquinas virtuales (VM de estilo clásico) o controladores de interfaces de red (NIC) individuales conectados a máquinas virtuales (VM tipo Resource Manager). Cuando un NSG se asocia con una subred, las reglas se aplican a todos los recursos conectados a la subred. Puedes restringir incluso más el tráfico si asocias un NSG a una máquina virtual o NIC.

Zonas privadas DNS de Azure

El DNS es responsable de traducir (o resolver) un nombre de servicio a su dirección IP. Azure DNS es un servicio de hosting para dominios DNS que proporciona resolución de nombres usando la infraestructura de Azure. Además de los dominios DNS con conexión a Internet, Azure DNS ahora es compatible con dominios DNS privados como una función de preview con las zonas privadas de Azure DNS. Los beneficios de seguridad de las zonas DNS privadas incluyen la capacidad de crear una infraestructura DNS dividida. Esto te permite crear zonas DNS privadas y públicas con los mismos nombres sin exponer los nombres internos. Además, el uso de las zonas DNS privadas de Azure elimina la necesidad de utilizar soluciones DNS personalizadas que podrían aumentar la superficie de vulnerabilidad con requisitos independientes de administración y actualizaciones.

// Más información: [Más información sobre las zonas DNS privadas](#)

VPN transversales

Azure es compatible con dos tipos de conexiones VPN transversales: VPN P2S y VPN S2S. Una conexión VPN P2S (punto a sitio) te permite crear una conexión segura a tu red virtual desde un equipo de cliente individual. Este tipo de conexión se establece desde el equipo cliente, lo cual es útil para los trabajadores remotos que quieren conectarse a una red virtual de Azure desde cualquier ubicación. La VPN P2S también es útil cuando solo tienes unos pocos clientes que necesitan conectarse a una red virtual. Por el contrario, se utiliza una conexión VPN S2S (sitio a sitio) para conectar tu red on-premises a una red virtual Azure sobre un túnel VPN IPsec/IKE (IKEv1 o IKEv2). Este tipo de conexión requiere un dispositivo VPN situado on-premises que tenga una dirección IP pública al exterior.

// Más información: [Más información sobre las VPN P2S y S2S.](#)

Azure ExpressRoute

[Azure ExpressRoute](#) te permite llevar tus redes on-premises al cloud de Azure a través de una conexión privada facilitada por un proveedor de conectividad sin tener que atravesar Internet. Con ExpressRoute, puedes establecer conexiones con los servicios de cloud de Microsoft, como Azure, Office 365 y Dynamics 365.

Azure Load Balancer

Puedes utilizar los equilibradores de carga para aumentar la disponibilidad de aplicaciones. Azure es compatible con equilibradores de carga internos y externos, los cuales pueden utilizarse en una configuración interna o pública.

Además, puedes configurar equilibradores de carga para que sean compatibles con puertos de alta disponibilidad (HA), en casos donde una regla de puertos HA sea una variante de una regla de equilibrio de carga configurada en el equilibrador de carga interno estándar. Puedes proporcionar una sola regla para equilibrar la carga de todos los flujos de TCP y UDP que llegan a todos los puertos de un equilibrador de carga interno.

```
// Más información: Más información  
sobre el equilibrador de cargas  
y reglas de puertos de alta  
disponibilidad
```


Registro y supervisión

Azure Log Analytics

[Azure Log Analytics](#) te ayuda a recopilar y analizar datos generados por recursos en tus entornos de cloud y on-premises. Proporciona conocimiento en tiempo real mediante paneles personalizados y de búsqueda integrada para analizar millones de registros en todas las cargas de trabajo y servidores sin importar su ubicación física.

Azure Monitor

[Azure Monitor](#) permite la supervisión básica de los servicios de Azure al permitir la recopilación de métricas, registros de actividades y registros de diagnóstico. Las métricas recopiladas proporcionan estadísticas de rendimiento para diferentes recursos, incluido el sistema operativo asociado con una máquina virtual.

El registro de actividades mostrará cuándo se crean o modifican nuevos recursos. Puedes ver estos datos con uno de los exploradores en el portal de Azure y enviarlos a Azure Log Analytics para el análisis de tendencias y detallado, o puedes crear reglas de alerta que notificarán proactivamente los problemas críticos.

Registros de flujo de los grupos de seguridad de red de Azure

[Los registros de flujo de los grupos de seguridad de red \(NSG\) de Azure](#) son una característica de Network Watcher que permite ver información acerca del ingreso y egreso de tráfico de IP mediante un grupo de seguridad de red. Los registros de flujos pueden analizarse para obtener conocimiento sobre el tráfico de una red y los problemas de rendimiento y seguridad relacionados con dicho tráfico.

Aunque los registros de flujos se enfocan en NSG, no se muestran de la misma forma que otros registros y se almacenan únicamente dentro de una cuenta de almacenamiento.

Azure Monitor Application Insights

[Azure Monitor Application Insights](#) es un servicio extensible de administración del rendimiento de las aplicaciones (APM) para desarrolladores web que trabajan en múltiples plataformas. También incluye potentes herramientas de análisis para diagnosticar problemas y entender qué hacen realmente los usuarios con tu aplicación. Funciona para aplicaciones en diversas plataformas hospedadas en el cloud y on-premises, incluidas .NET, Node.js y J2EE.

Application Insights se integra con tus procesos de DevOps y dispone de puntos de conexión con distintas herramientas de desarrollo. Puede supervisar y analizar telemetría para apps móviles al integrarse con Visual Studio App Center.

Proyecto de seguridad y cumplimiento en Azure

El proyecto de seguridad y cumplimiento de Azure: IA y datos médicos HIPAA/HITRUST proporciona herramientas y orientación para ayudar a implementar un entorno PaaS que cumpla con los estándares de la Health Insurance Portability and Accountability Act (La Ley de Portabilidad y Responsabilidad de Seguros de Salud, HIPAA) y la Health Information Trust Alliance (HITRUST).

Esta oferta PaaS permite adquirir, almacenar y analizar registros médicos personales y no personales, además de interactuar con ellos, en un entorno seguro de cloud de varios niveles implementado como una solución integral. El plan muestra una arquitectura de referencia en común que podría aplicarse a otros casos de uso además del cuidado de la salud, y está diseñado para simplificar la adopción de Azure.

// Más información: Más información sobre el proyecto de seguridad y cumplimiento de Azure

Documentación técnica y arquitectónica de seguridad de Azure

Azure cuenta con una enorme biblioteca de documentación técnica sobre seguridad que complementa la información de seguridad incluida con los servicios individuales. Se incluyen informes técnicos, documentos sobre prácticas recomendadas y listas de verificación en la página de información de Azure Security.

También se tratan temas de seguridad de cloud público en distintas áreas, tales como seguridad de red, seguridad de almacenamiento, seguridad de cómputo, administración de acceso e identidades, registro y auditoría, protección de carga de trabajo en el cloud, seguridad PaaS y más.

// Más información: Más información sobre la página de información Azure Security

Lectura adicional

Más información sobre Azure Security en los siguientes e-books gratuitos:

// Estrategia de cloud para la empresa

// Azure para arquitectos

04 /

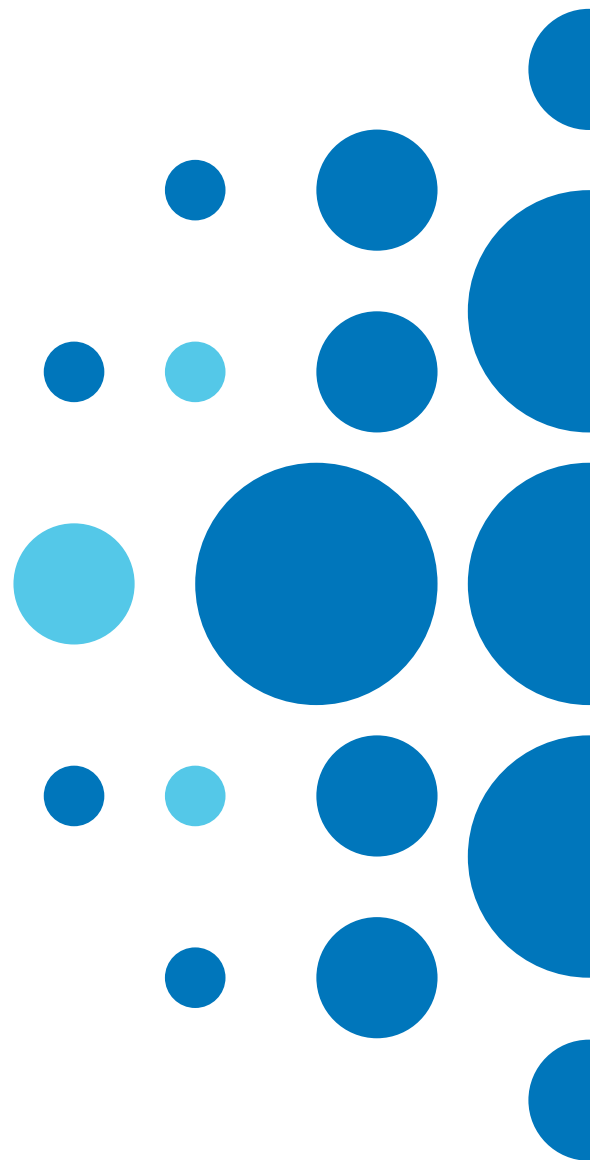
Agregar inteligencia a tu aplicación

¿Cómo puede Azure integrar la IA en tu aplicación?

La IA puede hacer que tu aplicación aventaje a las de la competencia. Imagina todo lo que podrías crear: aplicaciones que traduzcan el habla en tiempo real mientras hablas o una aplicación que te ayude a identificar partes de un motor en una sesión formativa con realidad mixta. Las posibilidades son infinitas. Sin embargo, la creación de una aplicación basada en IA parece difícil. ¿Cómo se crea un algoritmo que pueda comprender el habla, por ejemplo?

Por suerte, no tienes por qué hacerlo todo tú mismo. Azure está aquí para ayudarte. Ofrece muchos servicios de IA que puedes insertar en tu aplicación, como el de [Traducción de voz](#) (un servicio de [Azure Cognitive Services](#)) que traduce el habla en tiempo real.

Al utilizar un servicio como este, puedes consumir únicamente IA. Sin embargo, también puedes crear tus propios algoritmos de machine learning con servicios como [Azure Machine Learning Studio](#) y [el servicio Azure Machine Learning](#).



Qué utilizar y cuándo

Antes de analizar las opciones de IA en Azure, echemos un vistazo a la Tabla 4-1, en la que resumimos qué servicios están disponibles y cuáles son sus capacidades.

Tabla 4-1

	Búsqueda de Azure*	Azure Cognitive Services*	Azure Bot Service*	Azure Machine Learning Studio*	Azure Machine Learning*	Azure Spatial Anchors*	Azure Remote Rendering*
Mover datos de almacén a almacén	●						
Transformar datos	●	●	●	●	●	●	●
Consultar y filtrar datos de streaming				●		●	●
Proporcionar un modelo semántico in-memory para los usuarios		●			●		●
Los usuarios pueden consultar datos y crear paneles					●		
Analizar datos para uso posterior			●		●		●

* Los servicios con un asterisco tienen un nivel gratuito que puedes utilizar para empezar gratis.

Búsqueda de Azure

[La búsqueda de Azure](#) es una característica habitual en la mayoría de las aplicaciones, y sin embargo siempre ha sido una función difícil de implementar. La 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. Esto significa, por ejemplo, que podrías implementar la búsqueda de Azure fácilmente para ayudar a los usuarios a realizar búsquedas en tu catálogo de productos en una aplicación de comercio electrónico.

Hay muchas opciones para adaptar la búsqueda de Azure y hay muchas estupendas características que facilitarán las búsquedas a los usuarios, como:

- 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 Apache 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, el género, los sustantivos con plurales irregulares, la descomposición de palabras y la división de palabras para los idiomas sin espacios.
- Supervisión y creación de informes que facilitan información sobre 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.
- [Búsqueda cognitiva](#), que es un enfoque de prioridad de IA a la comprensión del contenido. La búsqueda cognitiva se basa en Búsqueda de Azure con Cognitive Services incorporado. Extrae datos desde casi cualquier fuente y aplica un conjunto de habilidades cognitivas modulares que extraen conocimiento. Este conocimiento se organiza y almacena en un índice, lo cual permite nuevas experiencias para explorar los datos mediante la búsqueda de Azure.

Las compañías petroleras y de gas recurren a la búsqueda cognitiva, ya que sus equipos de geólogos y otros especialistas necesitan comprender los datos geológicos y sísmicos. Estos equipos suelen haber recopilado durante décadas documentos PDF con fotos de muestras llenas de notas de campo escritas a mano. Los equipos necesitan relacionar lugares, expertos en el sector y eventos y luego explorar toda esta información para tomar decisiones claves.

La búsqueda cognitiva utiliza Cognitive Services para analizar todos estos datos, extraer información y correlacionarla, todo sin necesidad de escribir complicado software de reconocimiento de imágenes u OCR (reconocimiento óptico de caracteres).

```
// Pruébalo: Crea tu primer índice de la búsqueda de Azure en el portal
```

Cognitive Services

[Cognitive Services](#) ofrece algoritmos de machine learning creados por Microsoft y datos como servicio. Para la mayoría de los servicios, Microsoft también proporciona los datos para entrenar dichos algoritmos.

Para algunos servicios, puedes utilizar tus propios datos personalizados para entrenar los algoritmos.

Cognitive Services proporciona una manera excepcionalmente fácil de incorporar machine learning e inteligencia artificial en tu aplicación: simplemente invocando [API](#).

En la Tabla 4-2 se muestra qué API están disponibles actualmente. Ten en cuenta que la lista sigue aumentando.

Todos los servicios tienen un nivel gratuito que puedes utilizar para empezar.

Cada categoría de la tabla contiene diversos servicios que puedes usar al invocar una API. Algunas categorías contienen servicios personalizados, como Custom Vision, Language Understanding y Bing Custom Search. Estos servicios personalizados ofrecen algoritmos preconfigurados de machine learning, al igual que los demás servicios, y también te permiten utilizar tus propios datos para entrenar el modelo. Además de estos servicios, puedes utilizar los servicios en los [laboratorios de Cognitive Services](#). Los laboratorios contienen servicios experimentales que Microsoft está probando para ver si encajan bien con los casos de uso de los clientes. Uno de esos servicios experimentales es [Project Gesture](#), que permite detectar gestos, como un movimiento de la mano, e integrarlos en tu experiencia de usuario.

Veamos con más detenimiento algunos de los Cognitive Services.

// Pruébalo: Explora Cognitive Services

Tabla 4-2

Visión	Habla	Lenguaje	Conocimiento	Búsqueda
Computer Vision Face Video Indexer Content Moderator Custom Vision	Voz a texto Texto a voz Traducción de voz Speaker Recognition	Análisis de texto Traducción de texto Bing Spell Check Content Moderator Comprensión del lenguaje	QnA Maker	Bing Web Search Bing Visual Search Bing Entity Search Bing News Search Bing Custom Search Bing Image Search Bing Autosuggest Bing Video Search Bing Local Business Search

* Todos los servicios tienen un nivel gratuito que puedes utilizar para empezar.

Language Understanding

Utiliza el servicio [Language Understanding \(LUIS\)](#) para comprender lo que te dicen los usuarios en redes sociales, chatbots o aplicaciones con capacidades de voz. Por ejemplo, puedes reservar vuelos o programar reuniones.

Para utilizar el servicio Language Understanding, puedes darle ejemplos de lo que quieres que entienda, como “Reserva un vuelo a Seattle” o “Programa una reunión a la una con Bob”, y dile qué palabras buscas. En estos ejemplos, podrías estar buscando el destino del vuelo (Seattle) y la hora y la persona para la reunión (la una y Bob).

Después de que el servicio Language Understanding cree un modelo de machine learning basado en los ejemplos facilitados, podrá extraer información del lenguaje natural que aporten los usuarios.

```
// Pruébalo: Crea una nueva
  aplicación en el portal
  de LUIS
```

Custom Vision

Con el [servicio Custom Vision](#), puedes detectar información en imágenes sobre la base de tus propios datos de entrenamiento. Custom Vision funciona de forma similar a otros Cognitive Services en el sentido en que incluye un algoritmo de machine learning predefinido. Lo único que debes hacer es alimentar el servicio con tus datos.

Pongamos que quieres crear un modelo que pueda detectar los tipos de nubes de lluvia en el cielo, como cúmulos y estratos. Para crear este modelo, subes imágenes de diferentes tipos de nubes al portal de Custom Vision y les pones etiquetas, que le dicen al servicio cómo entrenar al modelo. En este ejemplo, etiquetarías una imagen con “cúmulo” o “estrato”.

Puedes entrenar el modelo después de cargar suficientes imágenes. Cuantas más imágenes etiquetadas cargues y cuanto más entrenamiento realices, el modelo será más preciso.

Cuando tengas un modelo que funcione correctamente, podrás empezar a usarlo invocando la API Custom Vision e introduciendo nuevas imágenes. Cuando cargas una nueva imagen, el servicio te dirá si la reconoce en función de las imágenes ya cargadas.

La Figura 4-1 muestra un ejemplo del aspecto del punto de conexión de la API.

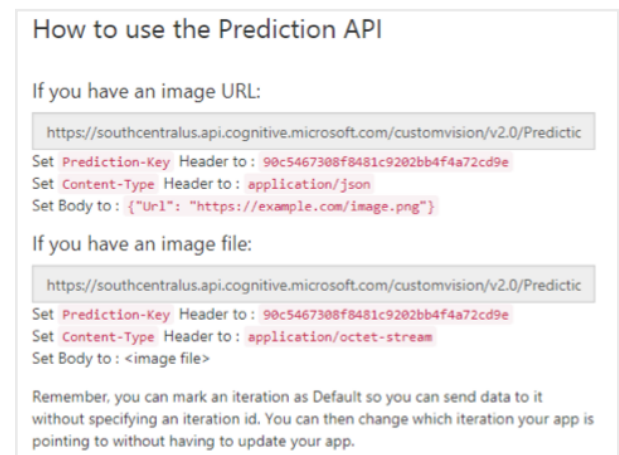


Figura 4-1

Utilizar el servicio Custom Vision para detectar información sobre la base de tu propio modelo es impresionante, pero Custom Vision puede hacer mucho más. El modelo que creas cuando entrenas el servicio Custom Vision con tus datos puede implementarse en el “sistema terminal inteligente”. Esto significa que el modelo y la API pueden ejecutarse en otro lugar que no sea el cloud, como un servidor on-premises dentro de un contenedor Docker o en otro dispositivo, como tu teléfono. Esto otorga gran flexibilidad porque no necesitas una conexión activa a Internet para utilizar las

capacidades del servicio Custom Vision. También puedes ejecutarlo localmente para lograr el mejor rendimiento. Además, el modelo que se ejecuta en el terminal inteligente no es demasiado grande: solo ocupa unas decenas de megabytes, dado que solo implementas el modelo y la API, no los datos de entrenamiento.

```
// Pruébalo: Crea tu propio  
proyecto de Custom Vision
```

Video Indexer

El servicio [Video Indexer](#) analiza los archivos de vídeo y audio que cargas. Este Cognitive Service también es parte del [conjunto de aplicaciones Media Analytics](#) de [Azure Media Services](#). Ofrece un algoritmo de machine learning predefinido y tú proporcionas los datos.

Entre [muchas otras](#), Video Indexer tiene las siguientes capacidades:

- Crea una transcripción del texto en un vídeo. Puedes refinar la transcripción manual y entrenar a Video Indexer para reconocer términos del sector, como "DevOps".
- Reconoce rostros e identifica quién aparece en un vídeo y en qué momentos. Video Indexer tiene la misma capacidad para el audio, en el que reconoce quién habla y cuándo.
- Reconoce texto visible en un vídeo (por ejemplo, el texto en una diapositiva) y lo transcribe.

- Realiza análisis de intención, que identifica cuándo se dice o se muestra algo positivo, negativo o neutro.

Como muestra la variedad de estas funcionalidades, Video Indexer combina muchos Cognitive Services, como [convertir voz en texto](#) y [Speaker Recognition](#). En conjunto, estos servicios ofrecen poderosas capacidades para que tu contenido sea más valioso, más accesible y puedas encontrarlo con mayor facilidad.

Puedes cargar archivos multimedia a Video Indexer mediante el portal de Video Indexer o la API. La Figura 4-2 muestra los resultados de un [vídeo Azure Friday](#) que se cargó al servicio Video Indexer.

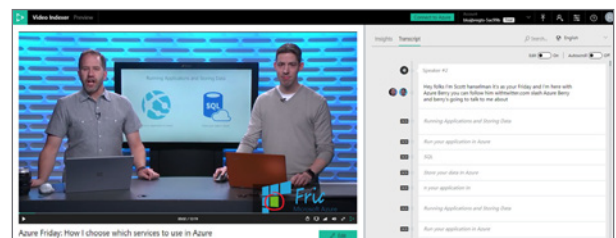


Figura 4-2

Como se muestra en la imagen, Video Indexer crea una transcripción del audio del vídeo. Puedes editar la transcripción e incluso traducirla a otros idiomas. También puedes ver el texto que ha reconocido Video Indexer en la diapositiva por detrás de los altavoces, marcado como "OCR". Puedes saltar a ese texto haciendo clic en él. Video Indexer ofrece esta funcionalidad para aplicaciones individuales integrando el [widget de Cognitive Insights](#).

```
// Pruébalo: Carga tu primer vídeo  
a Video Indexer
```

QnA Maker

El [Cognitive Service QnA Maker](#) ofrece una forma fácil de crear una capa de conversación sobre los datos existentes, como las páginas de preguntas frecuentes (FAQ), sitios web de soporte y manuales de productos. QnA Maker te ayuda a analizar y extraer la información y convertirla en pares de preguntas y respuestas que se pueden gestionar fácilmente. En términos sencillos, QnA Maker te permite crear aplicaciones que pueden facilitar información a tus usuarios en formato de conversación.

Con QnA Maker, es posible crear y administrar bases de conocimiento con el [QnA Maker Portal](#) fácil de usar o mediante API REST. Hemos simplificado el proceso de creación de bots para permitirte crear fácilmente un bot a partir de tu base de conocimiento, sin necesitar código ni cambios de configuración. Puedes ver aquí más información: [Crear un bot de QnA](#). Por supuesto que puedes utilizar QnA Maker para crear un bot mediante [Azure Bot Service](#) y complementar tu bot de QnA añadiendo el [Language Understanding Service](#). Para añadirle personalidad, puedes [añadir charla](#) a tu bot para que responda a escenarios de conversación informal “out of the box”.

Solo [pagas por el hosting de QnA Maker](#), no por cuántas veces los usuarios consultan el servicio resultante.

```
// Pruébalo: Crea tu servicio  
de QnA Maker basado en el  
conocimiento
```

Bing Autosuggest

[Bing Autosuggest](#) proporciona sugerencias de búsqueda mientras escribes. Esto permite ofrecer a tus usuarios una experiencia de búsqueda similar a las de Bing o Google, con la que los resultados de las búsquedas se automatizan o se completan.

Puedes introducir el texto de búsqueda, carácter por carácter, al servicio Bing Autosuggest, y rápidamente produce sugerencias de búsqueda en formato JSON.

Por ejemplo, cuando escribes el texto de consulta “Qué debo buscar”, el servicio produce el siguiente JSON (consulta la página 59 para ver un gráfico de mayor tamaño):

```
{  
  "type": "Suggestions",  
  "instrumentation": null,  
  "queryContext": {  
    "originalQuery": "what should I search for"  
  },  
  "suggestionGroups": [  
    {  
      "name": "Web",  
      "searchSuggestions": [  
        {  
          "url": "https://www.bing.com/search?q=what+should+I+search+for&FORM=USBAPI",  
          "urlPingSafe": null,  
          "displayText": "what should I search for",  
          "query": "what should I search for",  
          "searchKind": "WebSearch"  
        },  
        {  
          "url": "https://www.bing.com/search?q=what+should+I+search+for+on+bing&FORM=USBAPI",  
          "urlPingSafe": null,  
          "displayText": "what should I search for on bing",  
          "query": "what should I search for on bing",  
          "searchKind": "WebSearch"  
        },  
        {  
          "url": "https://www.bing.com/search?q=what+should+I+search+for+on+the+internet&FORM=USBAPI",  
          "urlPingSafe": null,  
          "displayText": "what should I search for on the internet",  
          "query": "what should I search for on the internet",  
          "searchKind": "WebSearch"  
        },  
        {  
          "url": "https://www.bing.com/search?q=what+should+I+search+for+today&FORM=USBAPI",  
          "urlPingSafe": null,  
          "displayText": "what should I search for today",  
          "query": "what should I search for today",  
          "searchKind": "WebSearch"  
        },  
        {  
          "url": "https://www.bing.com/search?q=what+should+I+search+for+in+dna+raw+data&FORM=USBAPI",  
          "urlPingSafe": null,  
          "displayText": "what should I search for in dna raw data",  
          "query": "what should I search for in dna raw data",  
          "searchKind": "WebSearch"  
        }  
      ]  
    }  
  ]  
}
```

Contiene todas las sugerencias. La consulta de búsqueda original está en la parte superior de los resultados.

```
// Pruébalo: Obtén una clave de API  
y prueba Bing Autosuggest gratis
```

```
{
  "_type": "Suggestions",
  "instrumentation": null,
  "queryContext": {
    "originalQuery": "what should I search for"
  },
  "suggestionGroups": [
    {
      "name": "Web",
      "searchSuggestions": [
        {
          "url": "https://www.bing.com/search?q=what+should+i+search+for&FORM=USBAPI",
          "urlPingSuffix": null,
          "displayText": "what should i search for",
          "query": "what should i search for",
          "searchKind": "WebSearch"
        },
        {
          "url": "https://www.bing.com/search?q=what+should+i+search+for+on+bing&FORM=USBAPI",
          "urlPingSuffix": null,
          "displayText": "what should i search for on bing",
          "query": "what should i search for on bing",
          "searchKind": "WebSearch"
        },
        {
          "url": "https://www.bing.com/search?q=what+should+i+search+for+on+the+internet&FORM=USBAPI",
          "urlPingSuffix": null,
          "displayText": "what should i search for on the internet",
          "query": "what should i search for on the internet",
          "searchKind": "WebSearch"
        },
        {
          "url": "https://www.bing.com/search?q=what+should+i+search+for+today&FORM=USBAPI",
          "urlPingSuffix": null,
          "displayText": "what should i search for today",
          "query": "what should i search for today",
          "searchKind": "WebSearch"
        },
        {
          "url": "https://www.bing.com/search?q=what+should+i+search+for+in+dna+raw+data&FORM=USBAPI",
          "urlPingSuffix": null,
          "displayText": "what should i search for in dna raw data",
          "query": "what should i search for in dna raw data",
          "searchKind": "WebSearch"
        }
      ]
    }
  ]
}
```

Azure Bot Service

La creación de un bot (una aplicación que interactúa con los usuarios de forma automática y autónoma) no es una tarea fácil. Necesitas hacer un seguimiento del contexto de las interacciones con cada usuario y debes estar preparado para responder ante muchos parámetros de interacción posibles.

[Azure Bot Service](#) te permite crear bots y experiencias inteligentes de nivel empresarial que pueden ayudarte a ampliar tu marca y mantener el control de tus datos. Puedes empezar con un bot sencillo de preguntas y respuestas o crear un asistente virtual sofisticado. Utiliza SDK y herramientas muy completas y de código abierto para conectar con facilidad tu bot entre distintos canales y dispositivos populares. Dale a tu bot la capacidad de hablar, escuchar y comprender a tus usuarios con una integración nativa con Cognitive Services.

Azure Bot Service facilita la creación de un bot y proporciona las siguientes funciones:

- Aporta una forma de hospedar y administrar los bots que hayas creado mediante el [Microsoft Bot Framework](#), con una gama completa de [SDK y herramientas de código abierto](#) para el desarrollo de bots.
- Se integra de forma nativa con Cognitive Services.
- Te ayuda a conectar tu bot con la ubicación de tus clientes, con conectores a canales como Facebook, Slack, Microsoft Teams, Line, Telegram y mucho más.
- Ofrece toda las ventajas de un servicio administrado en Azure, incluida una gran escalabilidad y entrega continua (CD) integradas, además de que pagas solo por lo que usas.

Por ejemplo, con Azure Bot Service puedes crear un bot que proporcione a los usuarios respuestas a sus preguntas más frecuentes.

Puedes usarlo con el [Cognitive Service QnA Maker](#). La interfaz del bot puede ser un cuadro de chat que esté en tu sitio web. También puedes crear un asistente virtual personalizado para tu marca con el [acelerador de soluciones para asistentes virtuales](#).

```
// Pruébalo: Empieza a trabajar  
con chatbots usando Azure  
Bot Service
```

Azure Machine Learning Studio

Puedes agregar increíbles funciones inteligentes a tus aplicaciones con servicios de Azure como Cognitive Services. Se basan en algoritmos de machine learning que Microsoft creó para usarlos como servicio. Sin embargo, hay otras maneras de utilizar machine learning en las aplicaciones. Primero hablemos de qué es el machine learning.

¿Qué es el machine learning?

El machine learning suele confundirse con la inteligencia artificial (IA), pero en realidad no son lo mismo. La IA se refiere a máquinas que realizan tareas que son características de la inteligencia humana. La IA también se puede implementar mediante el uso de machine learning, además de otras técnicas.

El machine learning es un campo de la ciencia computacional que da a los equipos informáticos la capacidad de aprender sin recibir programación explícita. El machine learning puede conseguirse mediante una o varias tecnologías algorítmicas, como redes neuronales, aprendizaje profundo o redes bayesianas.

Entonces, ¿qué supone el machine learning? La Figura 4-3 muestra el flujo de trabajo básico para el uso del machine learning.

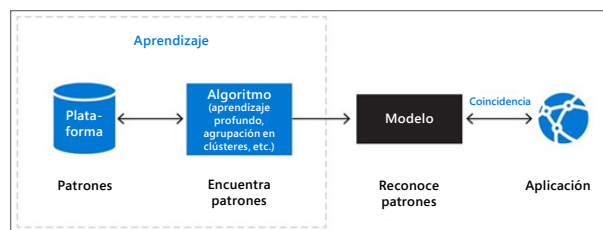


Figura 4-3

El proceso del machine learning funciona de la siguiente manera:

- Los datos contienen patrones. Probablemente ya conozcas algunos de los patrones, como los hábitos de compra de los usuarios. También es probable que haya muchos patrones en los datos que no te resulten familiares.
- El algoritmo de machine learning es la parte inteligente del software que te permite encontrar esos patrones en los datos. Este algoritmo puede ser uno que crees mediante técnicas como el aprendizaje profundo o el aprendizaje supervisado.
- Encontrar patrones en los datos utilizando el algoritmo de machine learning también se llama "entrenar un modelo de machine learning". Ese entrenamiento da como resultado un modelo de machine learning. Contiene los aprendizajes del algoritmo de machine learning.
- Las aplicaciones utilizan el modelo alimentando nuevos datos y trabajando con los resultados. Los nuevos datos se analizan según los patrones que se encuentran en los datos. Por ejemplo, cuando entrenas un modelo de machine learning para reconocer perros en imágenes, debería identificar un perro en una imagen que nunca haya visto.

La parte crucial de este proceso es que es un proceso iterativo. El modelo de machine learning se mejora constantemente al entrenarlo con nuevos datos y ajustando el algoritmo o ayudándole a diferenciar los resultados correctos de los incorrectos.

Uso de Azure Machine Learning Studio para crear modelos

Puedes utilizar [Azure Machine Learning Studio](#) para crear tus propios modelos de machine learning personalizados y exponerlos mediante servicios web para que tus aplicaciones puedan usarlos.

Machine Learning Studio es un servicio de Azure con el que puedes crear visualmente proyectos y experimentos de machine learning, vincular conjuntos de datos, crear notas y exponer modelos con servicios web. Studio es un portal que puedes usar desde tu navegador web y te permite crear algoritmos mediante un enfoque de arrastrar y soltar.

En Studio, puedes comenzar de cero o con uno de los muchos experimentos que se encuentran en la [galería](#), incluido uno para predecir la duración de las estancias en los hospitales y otro para detectar anomalías en streams de datos en tiempo real. Utiliza estos experimentos como base de un modelo de machine learning o para aprender sobre cómo pueden resolverse esos casos.

Un experimento de machine learning en Machine Learning Studio consta de varios pasos que manipulan datos y ejecutan algoritmos de machine learning en ellos. Puedes usar los pasos predefinidos disponibles en Studio para componer un algoritmo de machine learning.



Figura 4-4

En la Figura 4-4 se muestra un experimento en Machine Learning Studio, con el flujo de trabajo que se ejecuta para entrenar un modelo y las categorías de pasos predefinidos que pueden usarse en el flujo de trabajo.

Después de desarrollar tus experimentos y utilizarlos en tus datos para crear un modelo de machine learning, puedes publicarlos como servicios web. Cuando tus aplicaciones utilizan los servicios web, pueden enviar datos a tu modelo y recibir las predicciones de tu modelo.

// Pruébalo: Regístrate para utilizar
Azure Machine Learning Studio

Servicio Azure Machine Learning

Puedes utilizar el [servicio Azure Machine Learning](#) para crear algoritmos de análisis de datos con herramientas de código abierto como Python y la CLI de Azure. Al igual que con Machine Learning Studio, puedes crear el algoritmo que quieras, facilitando flexibilidad para diversos escenarios, como el análisis predictivo, las recomendaciones de datos y la clasificación de datos.

Con el servicio Azure Machine Learning, puedes crear algoritmos personalizados de machine learning desde cero. Esto es diferente de Machine Learning Studio, donde crearás visualmente un algoritmo conectando partes predefinidas de un algoritmo de machine learning. El servicio Azure Machine Learning es totalmente compatible con tecnologías de código abierto como Google [TensorFlow](#), [PyTorch](#) y [scikit-learn](#).

El servicio Azure Machine Learning es un servicio completo que ofrece capacidades de principio a fin. Puedes crear un algoritmo, preparar los datos, entrenar al algoritmo en función de los datos, probarlo, implementarlo y administrarlo y supervisarlos mientras se ejecuta. El servicio Azure Machine Learning funciona con

muchos servicios de Azure que pueden ayudar a crear, entrenar y ejecutar tu algoritmo. Por ejemplo, puedes crear tu algoritmo en Jupyter Notebook, entrenarlo con [Azure Databricks](#) e implementarlo en un clúster de contenedor de Kubernetes en [Azure Kubernetes Service](#).

```
// Introducción al servicio Azure  
Machine Learning con el portal  
de Azure
```

Herramientas para desarrolladores de inteligencia artificial

Herramientas de Visual Studio para IA

[Las herramientas de Visual Studio para IA](#) son una extensión gratuita de Visual Studio. Puedes utilizarlas para obtener acceso a diversos servicios y marcos de trabajo de IA, tales como [Microsoft Cognitive Toolkit \(CNTK\)](#), [TensorFlow](#), [Keras](#) y [Caffe2](#).

Las herramientas de Visual Studio Tools para IA te permiten crear algoritmos de machine learning de manera similar a Azure Machine Learning Studio. Puedes utilizar lenguajes como Python, C, C++ y C#, o puedes aprovechar una de las muchas muestras en la [galería](#) de experimentos de machine learning.

Con las Visual Studio Tools para IA puedes crear elementos de machine learning desde Visual Studio y aprovechar el potencial de Visual Studio para depurar algoritmos de machine learning y entrenar los modelos de machine learning. Desde Visual

```
// Pruébalo: Descarga la extensión  
de herramientas de Visual  
Studio para IA
```

Studio, puedes crear tareas de entrenamiento que puedan escalarse a muchas máquinas virtuales en Azure. También puedes supervisar el rendimiento del entrenamiento y luego generar un servicio web para utilizar el modelo de machine learning en tus aplicaciones. Puedes hacer todo esto sin salir de Visual Studio.

Kit de herramientas de IA para Azure IoT Edge

Al utilizar modelos de machine learning de forma local en los dispositivos (el perímetro inteligente) se consigue una ventaja muy importante: te permite usar el potencial de procesamiento del dispositivo sin depender de la conexión a Internet ni sufrir la latencia de una llamada al servicio web para obtener tus resultados.

Ya descrito antes en la sección sobre [Cognitive Services](#), el [servicio Custom Vision](#) ya puede ejecutarse en el perímetro. Cabe esperar que en el futuro se puedan ejecutar más servicios en el perímetro.

Para ejecutar modelos de machine learning en el perímetro, necesitamos las herramientas para ayudarte a implementar los modelos y los servicios web. El kit de herramientas de IA para Azure IoT Edge contribuye a este conjunto de herramientas al permitirte crear paquetes de modelos de machine learning con contenedores Docker compatibles con Azure IoT Edge y a exponer esos modelos como API REST.

El kit de herramientas de IA para Azure IoT Edge contiene ejemplos para ayudarte a comenzar, es de código abierto y está [disponible en GitHub](#).

IA y realidad mixta

Tus aplicaciones ya no están limitadas a un entorno 2D. El mundo es un lienzo para tus aplicaciones, que pueden verse respaldadas por la inteligencia espacial que facilitan los sensores del IoT, la realidad mixta y la visión informática. Con los servicios de realidad mixta de Azure, puedes dar vida a tus datos en 3D donde y cuando lo necesiten tus usuarios.

Azure Spatial Anchors

En el mundo de la realidad mixta, puedes integrar la información digital en el contexto de tu entorno físico, como un holograma de tus personajes de videojuego favoritos en la encimera de tu cocina. Con [Azure Spatial Anchors](#), puedes colocar contenido digital en un entorno físico y compartirlo con los usuarios mediante los dispositivos y plataformas que elijas.

Por ejemplo, cuando la gente entra en un hospital suele tener dificultades para encontrar el lugar al que quieren ir. Con Azure Spatial Anchors, el hospital puede crear una aplicación móvil que muestra información digital en el entorno físico del hospital para guiar a los usuarios a diversas ubicaciones. En la aplicación de su dispositivo iOS, los usuarios pueden utilizar las flechas direccionales de los paneles físicos informativos del hospital para llegar a su destino.

Otra forma en la que se está usando Azure Spatial Anchors es en una aplicación formativa para enfermeros. [Pearson Education](#) ha permitido a los estudiantes y profesores de enfermería practicar diagnósticos y tratar a los pacientes en 3D antes de enfrentarse a la presión de un caso real. Los alumnos y los profesores pueden utilizar dispositivos HoloLens o móviles y tabletas con iOS o Android.

Azure Spatial Anchors te permite compartir información digital y hologramas posicionados en el mundo físico. Funciona con aplicaciones desarrolladas con Unity, ARKit, ARCore y la Plataforma universal de Windows (UWP), y se puede utilizar con un dispositivo HoloLens, dispositivos basados en iOS con ARKit y dispositivos basados en Android con ARCore.

Con Azure Spatial Anchors, puedes proteger fácilmente tus datos espaciales y ofrecer a los usuarios acceso mediante Azure Active Directory. También puedes integrar el almacenamiento, la IA, el análisis y los servicios del IoT en tu aplicación espacial.

```
// Empieza a utilizar Azure  
Spatial Anchors entre sesiones  
y dispositivos
```

Azure Remote Rendering

Al utilizar modelos 3D en escenarios como revisiones de diseños y planes de procedimientos médicos, necesitas que tengan el mayor nivel de detalle posible. Todos los detalles importan.

Muchas empresas utilizan modelos 3D complejos que contienen cientos de millones de polígonos y dispositivos de perímetro con un potencial de procesamiento de gráficos bajo o medio, que no son capaces de renderizarlos. Tradicionalmente, los desarrolladores han intentado solucionar este problema mediante una técnica llamada "diezmado". Esto hace que el modelo sea más sencillo, ya que elimina polígonos para que se pueda mostrar en esos dispositivos.

Sin embargo, esta pérdida de detalle sacrifica la información necesaria para tomar la decisión adecuada en muchas situaciones. Con [Azure Remote Rendering](#), los modelos 3D se renderizan en el cloud y se transmiten a los dispositivos en tiempo real, sin renunciar a la calidad visual.

Esto permite que mantengas la calidad original del modelo y que interactúes con el contenido en dispositivos del perímetro, como auriculares y teléfonos móviles, dejando intactos todos los detalles.

Usar eventos y mensajes en una aplicación

Las aplicaciones modernas distribuidas globalmente deben lidiar a menudo con grandes cantidades de mensajes, por lo que se deben diseñar pensando en la desvinculación y el escalado. Azure proporciona varios servicios para ayudar con la ingesta y el análisis de eventos, además de los patrones de mensajería. Estos servicios también son fundamentales para crear aplicaciones inteligentes que aprovechen la inteligencia artificial.

Azure Service Bus

En Azure, la base de la mensajería es el [Azure Service Bus](#). Service Bus engloba un conjunto de servicios que se usan para los patrones de mensajería. Los servicios más importantes son las colas y temas de Azure Service Bus.

```
// Introducción a las colas  
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 los pedidos. El servicio web tardará mucho tiempo en procesar completamente el pedido, quizás hasta cinco minutos.

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 Service Bus. A continuación, la aplicación web informa 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 ha procesado 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 la aplicación.

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. Existen 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.

Esta es la diferencia entre ambos:

- 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.

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 el almacenamiento de colas de Azure

Las colas de Service Bus y el almacenamiento de colas de Azure hacen básicamente lo mismo, pero existen diferencias, como se muestra en la Tabla 4-3.

Tabla 4-3

Colas de Azure Service Bus	Almacenamiento de colas de Azure
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

[Azure Event Hubs](#) puede ayudar a las empresas a capturar enormes cantidades de esos datos para analizarlos o transformarlos y moverlos para su uso posterior.

Event Hubs se diseñó para la ingesta masiva de datos. Gestiona sin esfuerzo millones de mensajes por segundo. Puedes retener mensajes hasta un máximo de siete días o 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.

// Pruébalo: Empieza ahora a enviar mensajes a Azure Event Hubs

Azure IoT Hub

Al igual que Event Hubs, [Azure IoT Hub](#) está diseñado para la ingesta masiva de datos. Está pensado especialmente para la gestión del enorme volumen de mensajes de datos procedentes de los dispositivos del Internet de las cosas (IoT), 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 puede enviarles mensajes a los dispositivos. Tiene la capacidad de administrar toda la infraestructura de IoT: puedes utilizarlo para registrar dispositivos, informar de su estado, administrarlos protegiéndolos y reiniciándolos y enviar datos a los mismos.

```
// Pruébalo: Conecta tu
dispositivo a tu IoT Hub
```

Azure Event Grid

[Azure Event Grid](#) ofrece un tipo de mensajería diferente: 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 obtener 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 cuando se agrega un nuevo Blob, por ejemplo, o cuando se agrega un nuevo usuario a una suscripción de Azure. Azure Event Grid detecta estos eventos y hace que estén disponibles para los controladores de eventos y los servicios que se han suscrito a los eventos, como se muestra en la Figura 4-5.

Los controladores de eventos pueden ser Azure Functions o Azure Logic Apps, que pueden actuar sobre los datos del evento.

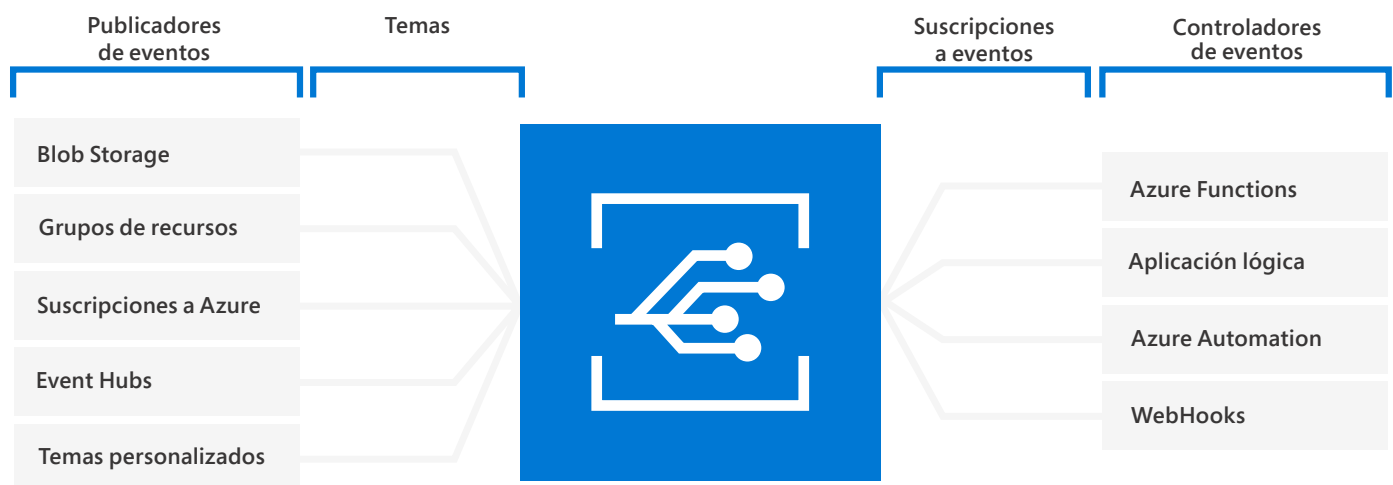


Figura 4-5

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 pagarás cuando se utilice.

Puedes utilizar Azure Event Grid si quieres recibir una notificación por correo electrónico cada vez que se agrega o elimina un usuario de tu lista de correos en 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. Azure Event Grid indica a continuación a Logic Apps que envíe un correo electrónico que contiene el nombre del usuario que se ha agregado o eliminado y la acción que se realizó.

```
// Pruébalo: Supervisa los cambios  
de las máquinas virtuales con  
Azure Event Grid y Logic Apps
```

Azure SignalR Service

Puedes utilizar Azure SignalR Service para añadir funcionalidades web en tiempo real a tus aplicaciones. El servicio se basa en ASP.NET Core SignalR y se ofrece como un servicio totalmente administrado y autónomo en Azure.

SignalR puede actualizar las aplicaciones conectadas en tiempo real mediante HTTP, sin necesidad de que las aplicaciones busquen actualizaciones o envíen nuevas solicitudes HTTP. Esto te permite crear experiencias web transparentes que actualizan la información sobre la marcha. Por ejemplo, una aplicación de subastas podría utilizar SignalR para actualizar la oferta más reciente tan pronto ocurra, sin necesidad de actualización toda la página ni tener que sondear por nueva información.

Hospedar un servidor de SignalR no es una tarea sencilla y puede ser difícil escalar y protegerlo adecuadamente. Cuando utilizas el servicio Azure SignalR completamente administrado, es fácil de configurar, y la seguridad, disponibilidad, rendimiento y escalabilidad son también completamente administrados.

```
// Pruébalo: Crea una sala de chat  
con SignalR
```

Qué utilizar y cuándo

Azure proporciona numerosas opciones para realizar mensajería y para desacoplar aplicaciones. ¿Cuál se debe utilizar y cuándo? La Tabla 4-4 resume las diferencias para ayudarte a elegir.

Lectura adicional

Puedes obtener más información sobre el uso de los servicios de IA de Azure en tu aplicación en este e-book gratuito:

// [Guía del desarrollador para aplicaciones de IA](#)

Tabla 4-4

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

* Los servicios con un asterisco tienen un nivel gratuito que puedes utilizar para empezar gratis.

05 /

Conecta tu empresa con IoT

¿Cómo puede Azure conectar, proteger, gestionar, supervisar y controlar tus dispositivos en el cloud?

Una de las cosas más emocionantes que puedes hacer como desarrollador es causar impacto en el mundo real. Eso solo lo puedes conseguir con dispositivos como robots, sensores y microcontroladores. Al crear aplicaciones con estos dispositivos, puedes conseguir cosas como predecir cuándo necesitará mantenimiento una máquina antes de que haga falta repararla, o incluso crear un coche autónomo.

Trabajar con dispositivos para causar un impacto en el mundo real puede ser complejo si desarrollas tú mismo todo el software. Por suerte, Azure ofrece soluciones que pueden ayudarte a facilitar mucho todo esto. Por ejemplo, puedes usar [Azure IoT Hub](#) para introducir mensajes de forma segura a partir de los sensores y realizar mantenimiento de los dispositivos, como enviar mensajes a los dispositivos y reiniciarlos. Cuando creas algo con un microcontrolador (equipos informáticos de un único chip), plantéate usar [Azure Sphere](#), que es una combinación de hardware, SO y servicios en el cloud con seguridad integrada.

Analicemos los servicios de Azure que te pueden ayudar a crear aplicaciones del IoT increíbles.



Azure IoT Hub

En el corazón del IoT de Azure se encuentra [Azure IoT Hub](#), que es una PaaS de cloud abierta y flexible para conectar, supervisar y administrar miles de millones de dispositivos de forma segura y escalable. Ya hemos hablado de Azure IoT Hub en un capítulo anterior, pero hay mucho más cosas de qué hablar en el contexto de IoT.

Ya sabes que puedes utilizar IoT Hub para ingerir grandes cantidades de mensajes que por lo general provienen de dispositivos de IoT, como los mensajes que contienen datos de los sensores de temperatura. Pero además, IoT Hub es único, porque no solo recibe mensajes, también envía los comandos de vuelta a los dispositivos. Establece una comunicación bidireccional con los dispositivos e incluso te permite ejecutar código en ellos.

IoT Hub es extremadamente poderoso porque te permite administrar y controlar los dispositivos de diversas formas, como enviándoles un mensaje para que se reinicien o ejecutando un script de arranque. IoT Hub es el servicio fundamental que permite una aplicación IoT completa en Azure. Para ayudarte a aprovisionar dispositivos a escala, Azure proporciona el [servicio IoT Hub Device Provisioning](#).

Este servicio permite aprovisionamiento oportuno y sin intervención al IoT Hub apropiado sin necesidad de intervenir, lo cual te permite aprovisionar miles de millones de dispositivos de forma segura y escalable. El servicio te puede ayudar en muchas situaciones de aprovisionamiento, tales como conectar dispositivos a un IoT Hub y ejecutar sus scripts de configuración inicial, equilibrar cargas de dispositivos en distintos centros y reaprovisionar con base en un cambio de los dispositivos.

IoT Hub puede también conectarse a dispositivos que pueden ejecutar cargas de trabajo desarrolladas en el cloud, incluidas aquellas que ejecutan módulos y operaciones de Azure IoT Edge. Cuando un dispositivo está conectado a IoT Hub, el hub mantiene un registro de su identidad. Esto permite que IoT Hub envíe mensajes y supervise el dispositivo; también permite que IoT Hub proteja el dispositivo y las comunicaciones entre ellos. Los dispositivos pueden requerir autenticación de IoT Hub mediante varios protocolos de seguridad de prácticas recomendadas del sector, tales como certificados X.509 y autenticación basada en token SAS. Puedes administrar la seguridad de cada dispositivo conectado y revocar los privilegios si ya no quieres que un dispositivo en particular se conecte.

Cuando los dispositivos envían un mensaje a Azure IoT Hub, puedes almacenar el mensaje o enrutarlo a otro servicio para realizar análisis o una acción. Es posible, por ejemplo, derivar los mensajes entrantes mediante la función de desvío de mensajes de IoT que ofrece simplicidad, fiabilidad y escalabilidad sin la complejidad de crear soluciones de derivación personalizadas.

Otra opción para integrar eventos del IoT en los servicios o aplicaciones empresariales de Azure es usar Azure Event Grid, una solución de enrutamiento de eventos totalmente administrada que utiliza un modelo de publicación-suscripción. IoT Hub y Event Grid funcionan en conjunto para integrar eventos de IoT Hub en Azure y servicios ajenos a Azure en tiempo casi real.

También puedes crear túneles de comunicación bidireccional utilizando transmisiones de dispositivos. Las transmisiones de dispositivos de Azure IoT Hub facilitan la creación de túneles TCP bidireccionales seguros para una variedad de casos de comunicación entre el cloud y los dispositivos.

Ejemplo: Azure IoT Hub

Una empresa que ofrece información sobre el movimiento y uso de camiones quiere escalar su negocio. Anteriormente, la compañía solía controlar sus activos utilizando un código personalizado en una app nativa de teléfono que llamaba a un servicio web personalizado y sondeando los GPS físicos que se encuentran en los camiones. Esta solución tenía un mantenimiento complejo, porque resultaba complicado aprovisionar nuevos activos con nuevos dispositivos, y la empresa necesitaba incorporar a un nuevo cliente que tenía más de 2000 activos.

Ahora, la empresa usa Azure IoT Hub para la gestión de dispositivos y la comunicación. Puede utilizar

el servicio de aprovisionamiento de dispositivos de IoT Hub para incorporar los 2000 nuevos dispositivos y conectarlos a un IoT Hub específico para ese cliente. La aplicación telefónica ahora utiliza Node.js y el [Azure IoT Device SDK](#) para interactuar con IoT Hub. Es notable que ahora la compañía tiene control sobre la seguridad de sus dispositivos y puede detectar su estado y reiniciarlos en caso necesario. Además, la empresa enruta los datos de sus aparatos GPS mediante Azure Stream Analytics, así que solo conserva los datos de cambios en los GPS. Esto reduce la acumulación de datos, dado que los aparatos envían su ubicación cada segundo.

Al usar este servicio central único, Azure IoT Hub, le permitió a esta compañía escalar y madurar su negocio al proporcionar seguridad y administración de dispositivos de primer nivel. También abrió nuevas posibilidades para lograr hacer más con los dispositivos empresariales más de lo que nunca creyeron posible.

Azure IoT Central

[Azure IoT Central](#) es una plataforma hospedada de soluciones del IoT que te permite crear aplicaciones del IoT ricas con solo navegar con la ayuda de varios asistentes.

No es necesario crear código ni realizar ninguna configuración complicada: IoT Central se hace cargo de todo y aprovisiona y configura todo lo que necesites, incluido Azure IoT Hub.

Obtienes las mismas capacidades que obtendrías si hubieses creado la solución desde cero, pero sin la necesidad de contar con años de experiencia en programación. Si quieres tener más control sobre ciertas áreas, puedes profundizar en cualquier momento y ajustar la solución a tus necesidades.

Aceleradores de soluciones de Azure IoT

[Los aceleradores de soluciones de Azure IoT](#)

son un punto de partida excelente para desarrollar tu solución de IoT.

Estas plantillas completas y personalizables para escenarios de IoT habituales hacen de todo, desde supervisar los dispositivos y protegerlos hasta proporcionar una interfaz de usuario. También te ayudan a conectar dispositivos existentes y nuevos dispositivos. La figura 5-1 muestra un ejemplo de una plantilla de solución.

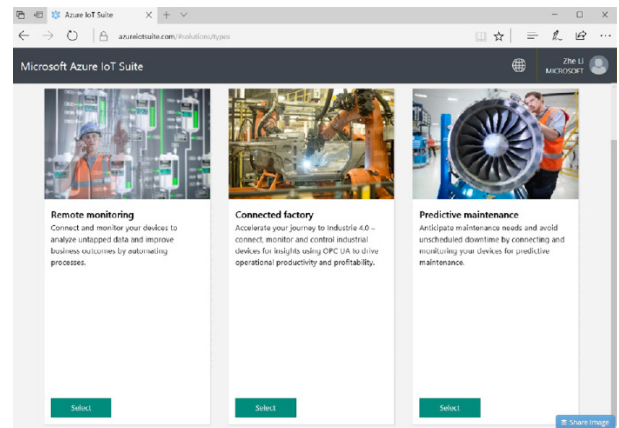


Figura 5-1

Conecta y supervisa tus dispositivos con control remoto. Obtén mayor visibilidad de tus dispositivos, activos y sensores, dondequiera que estos se encuentren. Recopila y analiza datos de los dispositivos en tiempo real con un acelerador de soluciones preconfigurado remoto que activa alertas y acciones automáticas, todo desde diagnósticos remotos hasta solicitudes de mantenimiento.

Hay muchos otros aceleradores de soluciones de IoT, como aquellos que mejoran las eficiencias industriales con una fábrica conectada, aumentan la fiabilidad del equipo con mantenimiento predictivo y desarrollan y prueban tu solución de IoT con simulación de dispositivos.

Azure IoT Edge

En las aplicaciones del IoT modernas, el procesamiento de datos puede ocurrir en el cloud o en los dispositivos. El procesamiento en los dispositivos se llama "computación periférica".

La computación periférica se utiliza cuando no quieres (o no puedes) depender de la conexión al cloud, cuando buscas mejorar el rendimiento de tu aplicación eliminando la conexión al cloud o cuando no puedes comunicarte con el cloud desde el dispositivo por razones de seguridad, privacidad o reglamentarias.

Para situaciones como estas, puedes utilizar [Azure IoT Edge](#). Azure IoT Edge se gestiona desde IoT Hub, lo que te permite trasladar partes de tus cargas de trabajo al perímetro. Esto reduce el tiempo que invierten los dispositivos en enviar mensajes al cloud y permite escenarios sin conexión y reacciones más veloces a los cambios de estado.

Azure IoT Edge se compone de los siguientes tres componentes:

- **Los módulos de IoT Edge** son contenedores que ejecutan servicios de Azure, servicios de terceros y tu propio código. Se implementan en dispositivos de IoT Edge y se ejecutan localmente en esos dispositivos.
- El **tiempo de ejecución de IoT Edge** se ejecuta en cada dispositivo de IoT Edge y administra los módulos implementados en cada dispositivo.
- **IoT Hub** expone interfaces específicas para supervisar y administrar de forma remota dispositivos IoT Edge disponibles mediante el portal de Azure, Azure CLI o SDK.

Estos tres componentes funcionan conjuntamente en los dispositivos y en el cloud para ejecutar tus cargas de trabajo en el perímetro inteligente.

Es posible ejecutar muchos servicios de Azure en la periferia para ayudarte en ciertas situaciones, y la lista de servicios disponibles sigue aumentando. La Tabla 5-1 enumera solo algunos de ellos.

Tabla 5-1

Sí quieres	Usa esto en Azure IoT Edge
Desarrollar e implementar modelos de inteligencia artificial	Machine Learning
Personalizar modelos de visión de computadora para el caso de uso	Custom Vision Service
Procesamiento de flujos de datos en tiempo real	Stream Analytics
Procesamiento de eventos con código sin servidor	Functions
Implementar una base de datos SQL Server en el perímetro	Base de datos SQL Server
Cumplir con los estándares de interoperabilidad de Industria 4.0	Arquitectura unificada de OPC
Desarrollar lógica personalizada	Módulo personalizado

Cuando comiences a utilizar Azure IoT Edge, podrás crear aplicaciones rápidas que ejecuten algoritmos de machine learning localmente y ofrezcan comentarios instantáneos sobre sus descubrimientos.

Azure Digital Twins

En el mundo del IoT, podrías trabajar con muchos dispositivos y sensores del IoT que funcionan en relación con las personas y los objetos. Para comprender realmente los datos que captan los dispositivos del IoT, necesitas comprender el entorno físico en el que funcionan los dispositivos.

Por ejemplo, si tienes un sensor de temperatura, los datos de temperatura por sí solos no facilitan demasiada información. Sin embargo, si sabes en qué estancia se encuentra el sensor, qué tamaño tiene esa estancia, qué otros dispositivos hay en la sala y qué datos capturan, o cuánta gente hay dentro y cómo se mueven, obtendrás un panorama mucho más significativo de lo que implican los datos de temperatura. Puedes modelar el entorno físico en el que residen tus dispositivos del IoT con

[Azure Digital Twins](#). Este servicio te permite crear un gráfico de datos que incluye lugares, personas y cosas: por ejemplo, un edificio de oficinas que contiene una sala con personas y sensores.

Con Azure Digital Twins, puedes facilitar contexto a los datos desde varios orígenes y relacionarlos entre sí. Por ejemplo, los datos de temperatura y humedad de sensores de la misma sala. Esto te permite consultar los datos en el contexto de un espacio, en lugar de a partir de sensores individuales.

Azure Digital Twins también te permite gestionar los permisos de los datos y los dispositivos en el contexto del entorno físico. Puedes usar Azure AD para especificar que determinados usuarios pueden acceder a los datos únicamente desde una ubicación física determinada.

```
// Empieza por encontrar salas  
disponibles con Azure Digital  
Twins.
```

Azure Sphere

Cada vez más dispositivos contienen microcontroladores (equipos informáticos de un único chip) para que sean más inteligentes. Esto permite que se den escenarios como el de una lavadora que envía una notificación de mantenimiento rutinaria al propietario.

Sin embargo, proteger los dispositivos con microcontroladores integrados es complejo, como ponen de manifiesto los numerosos incidentes de seguridad con dispositivos conectados que se han producido en los últimos años.

[Azure Sphere](#) ofrece una solución para proteger dispositivos basados en microcontroladores. La plataforma Azure Sphere consiste en un chip de microcontroladores seguro, un SO y un conjunto de servicios del cloud que se conectan con el microcontrolador y lo actualizan a medida que se ejecuta. Esta combinación supone la base de un mundo conectado y seguro.

Puedes utilizar las herramientas de desarrollo de Visual Studio para desarrollar aplicaciones que se ejecutan en Azure Sphere. Esto abre el mundo del desarrollo de microcontroladores a un grupo de desarrolladores mucho mayor.

// Más información sobre Azure Sphere

Obtener más información acerca de Azure IoT

Las soluciones de Azure IoT son fáciles de utilizar y hay muchos recursos relacionados, incluidos:

- [Azure IoT School](#): Esta academia online gratuita ofrece cursos integrales sobre Azure IoT, con una variedad de cursos desde nivel inicial hasta avanzado.
- [Creación de soluciones de IoT con Azure](#): Este curso online te guía a través de los principales conceptos de Azure IoT a tu propio ritmo.
- [Página de desarrollo de aplicaciones de Azure IoT](#): Este recurso te proporciona información general de los servicios de Azure IoT y ejemplos de cómo puedes utilizarlos.
- [Aceleradores de soluciones de Azure IoT](#): Puedes utilizar estas plantillas para comenzar fácilmente a utilizar los servicios de Azure IoT.
- [Azure IoT Hub](#): Este recurso te proporciona información general de Azure IoT Hub y ejemplos de cómo puedes utilizarlos.
- [Azure IoT Edge](#): Este recurso te proporciona información general de Azure IoT Edge y un ejemplo de cómo puedes utilizarlo.
- [Vídeos técnicos de Azure IoT](#) Más información sobre el IoT en Channel 9.

Qué utilizar y cuándo

Ahora que has leído acerca de los servicios de Azure IoT disponibles, ¿cómo sabes qué servicio debes utilizar para tu situación? La Tabla 5-2 muestra cuándo deberías utilizar cada opción del IoT en Azure.

Lectura adicional

Más información sobre el uso de Azure en tu solución del IoT en los siguientes e-books gratuitos:

// [Guía del IoT para desarrolladores](#)

// [La disrupción por diseño](#)

Tabla 5-2

	Azure IoT Hub	Azure IoT Central	Aceleradores de soluciones de Azure IoT	Azure IoT Edge
Crear una solución de IoT con mucho control y código personalizado	●			
Crear una solución de IoT sin preocuparte por el código ni la administración de los servicios de Azure		●		
Crear una solución de IoT para una situación común con mínima configuración y escritura de código			●	
Ejecutar localmente cargas de trabajo de inteligencia artificial en dispositivos de IoT	●*			●

Todos los servicios tienen un nivel gratuito que puedes utilizar para empezar.

*Necesitas Azure IoT Hub para administrar implementaciones y dispositivos de Azure IoT Edge.

06 /

Dónde y cómo implementar los servicios de Azure

¿Cómo puede Azure implementar tus servicios?

Azure tiene una opción para cada tipo de organización, incluidas aquellas que necesitan que Azure esté en su propio centro de datos. Puedes implementar tus aplicaciones en el cloud público de Azure u on-premises en [Azure Stack](#) y elegir qué nivel de portabilidad han de tener tus aplicaciones.

También es posible desarrollar aplicaciones en contenedores para implementarlas on-premises o en otro cloud, o utilizando [plantillas de Resource Manager](#) para desarrollar toda tu infraestructura como código.

Vamos a explorar estas opciones con mayor detalle.



Infraestructura como código

La infraestructura como código (IaC) captura definiciones del entorno como código declarativo, como los documentos JSON, para el aprovisionamiento y la configuración automatizadas. Todos los servicios de Azure presentados en esta guía se basan en [Azure Resource Manager](#), el cual puedes usar para documentar tu entorno como IaC, gracias a [las plantillas de Azure Resource Manager](#). Estas plantillas son archivos JSON que describen lo que se desea implementar y especifican los parámetros correspondientes.

Es fácil crear plantillas de Azure 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 Azure Resource Manager desde el portal de Azure haciendo clic en el botón Script de automatización, que está disponible en la barra de menús de todos los recursos disponibles en el portal de Azure. Esto crea la plantilla de Azure Resource Manager para el recurso dado e incluso código para crear el recurso utilizando la CLI de Azure, PowerShell, .NET y otros.

Una vez que tengas una plantilla de Azure Resource Manager, puedes implementarla en Azure usando PowerShell, la CLI de Azure o Visual Studio. También puedes automatizar tu implementación en un proceso de desarrollo continuo (CD) usando Azure DevOps.

Un gran ejemplo de implementación de los recursos en el cloud utilizando Azure Resource Manager es el [botón Implementar en Azure](#) que está en muchos repositorios de GitHub.

Además de utilizar Resource Manager para IaC, puedes incorporar las herramientas y habilidades que ya posees, tales como [Ansible](#), [Chef](#) y [Terraform](#) para aprovisionar y administrar la infraestructura de Azure directamente.

Azure Blueprints

Es fácil utilizar las plantillas de Azure Resource Manager, los grupos de recursos, las identidades de usuarios y los derechos y políticas de acceso para diseñar y crear la infraestructura completa. ¿Pero cómo se mantienen todas estas cosas en conjunto? ¿Y cómo se controla en qué entornos se ha implementado cada infraestructura y qué versión del artefacto está implementada ahora?

Organiza todos tus artefactos de infraestructura con [Azure Blueprints](#). Azure Blueprints facilita un mecanismo que te permite crear y actualizar artefactos, asignarlos a entornos y definir versiones. Puedes almacenar y administrar estos artefactos, además de administrar sus versiones y relacionarlas con entornos.

Esto te ayudará a organizar tu infraestructura y crear un contexto para las plantillas de Azure Resource Manager, las identidades de usuarios, los grupos de recursos y las políticas.

```
// Introducción a la definición  
y asignación de un Azure  
Blueprint en el portal  
de Azure.
```

Contenedores de Azure

El “uso de contenedores” es una de esas palabras tecnológicas de moda que aparecen en todas las noticias. Sin embargo, los contenedores no son solo una moda: 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 tradicional y, por tanto, es mucho más útil para desarrollar, probar y ejecutar aplicaciones en producción.

La mayor ventaja de los contenedores es que un contenedor es siempre igual. Un contenedor se ejecuta localmente al desarrollar la aplicación y después 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 que en producción. Con los contenedores, la excusa típica de los desarrolladores, “en mi equipo sí funciona”, ahora significa que también funcionará en producción.

Hay muchas tecnologías para ejecutar contenedores; una de ellas es [Docker](#). Azure puede ejecutar y administrar los contenedores con [Azure Container Instances](#) y [Azure Kubernetes Service](#). Además, puedes ejecutar contenedores en [Web App for Containers](#) y en [Azure Batch](#). La Tabla 6-1 muestra qué servicio podrías elegir en cada caso cuando trabajas con contenedores.

Tabla 6-1

	Azure Kubernetes Service	Azure Container Instances	Web App for Containers	Contenedores en Azure Batch
Para las implementaciones de producción de sistemas complejos (con un orquestador de contenedores)	●			
Para ejecutar configuraciones sencillas (posiblemente sin orquestador)		●	●	
Para cargas de trabajo con tiempos de ejecución largos en los contenedores	●			●
Para cargas de trabajo con tiempos de ejecución cortos en los contenedores		●		●
Para organizar un sistema basado en contenedores	●			
Coordinar con orquestadores de código abierto Kubernetes	●			
Organizar con orquestador integrado				
Con características de App Service como ranuras de implementación			●	

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 del portal de Azure y las mismas API, con las que puedes usar la CLI de Azure, 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, excepto que ahora se ejecuta on-premises. Si decides cambiar al cloud público, puedes simplemente transferir los servicios de Azure Stack a Azure.

Ejemplo: Azure Stack

Una empresa que ofrece vacaciones en crucero de lujo cuenta con varios productos de software para ayudar con las tareas de los cruceros, incluida una aplicación de gestión de camarotes y una de gestión de pasajeros. Todo el crucero depende de estas aplicaciones. Anteriormente, las aplicaciones todavía se ejecutaban en servidores a bordo de los cruceros. La empresa estaba obligada a hacerlo así porque los cruceros no tenían conexión a Internet durante todo el trayecto.

La empresa descubrió que ejecutar sus aplicaciones on-premises implicaba demasiada carga, dado que tenía que mantener las máquinas virtuales y sistemas operativos al mismo tiempo que debía lidiar con importantes problemas de disponibilidad.

Ahora, la empresa ejecuta sus aplicaciones en Azure Stack, que se ejecuta a bordo de los cruceros. Azure Stack ofrece los mismos servicios que Azure, así que la implementación y administración de aplicaciones es mucho más fácil. La compañía también utiliza Azure App Service para ejecutar sus aplicaciones, lo cual permite enfocarse en las aplicaciones en lugar de en mantener las máquinas virtuales y los sistemas operativos. Es más: los usuarios disfrutaron de la mayor disponibilidad que, al ser parte de Azure, es también parte de Azure Stack.

Dónde y cuándo implementar

Si quieres implementar servicios basados en IaaS (en los que tienes control del sistema operativo), considera estas opciones:

- **On-premises o en cualquier otro lugar** (como tu PC local u otro cloud), puedes utilizar:
 - Azure Stack (donde implementas servicios como máquinas virtuales)
 - Cualquiera de los servicios de contenedores de Azure (dado que los contenedores pueden ejecutarse en cualquier lugar)
- **En el cloud público de Azure**, puedes utilizar:
 - Contenedores (dado que los contenedores también pueden ejecutarse en cualquier servicio de contenedores de Azure)

Si quieres implementar servicios basados en PaaS (en los que tienes menos control, pero la plataforma realiza el trabajo pesado), considera las siguientes opciones:

- **On-premises o en cualquier otro lugar** (como tu PC local u otro cloud), puedes utilizar:
 - Azure Stack (dado que puedes implementar servicios PaaS como App Service en Azure Stack)
- **En el cloud público de Azure**, puedes utilizar:
 - Cualquier servicio Azure PaaS que definas como una plantilla en Resource Manager

Lectura adicional

Obtén más información sobre la implementación de tus aplicaciones en Azure y la reducción de costes en estos e-books gratuitos:

// Aspectos básicos de la migración al cloud

// Sacar el máximo partido al cloud en cualquier lugar

// DevOps efectivas

// Azure para arquitectos

07 /

Compartir
código,
controlar el
trabajo y enviar
el software

¿Cómo puede ayudarte Azure a planificar de forma más sensata, colaborar mejor y enviar tus aplicaciones más rápido?

Seguro que has pasado fines de semana y noches enteras implementando nuevas versiones de tus aplicaciones. Si es así, probablemente has dedicado también un montón de tiempo a arreglar los problemas que evitan que los usuarios accedan a esas nuevas versiones. Hay una forma mejor de hacerlo.

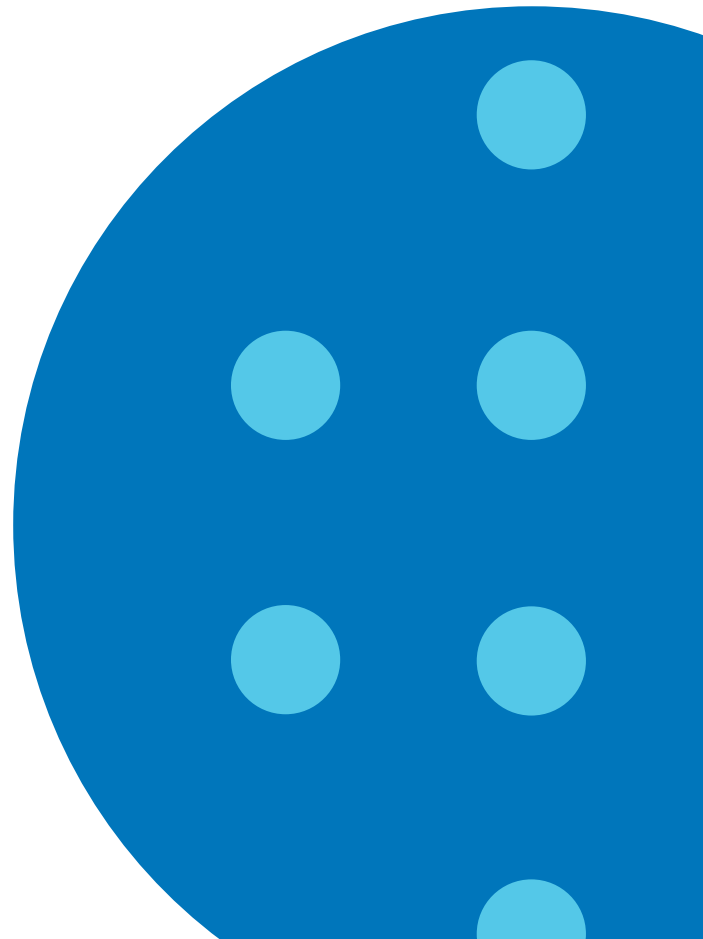
[Azure DevOps](#) es un conjunto de soluciones que pueden ayudar a automatizar tus compilaciones e implementaciones y comprobar tu código y tus aplicaciones automáticamente antes de su lanzamiento.

Para ayudarte a crear, implementar, probar y realizar un seguimiento de tu código y tus aplicaciones, Azure DevOps incluye:

- [Azure Boards](#)
Puedes usar Azure Boards para planificar, realizar un seguimiento y hablar sobre el trabajo entre diferentes equipos.
- [Azure Repos](#)
Puedes utilizar Azure Repos para colaborar en el desarrollo de código con repositorios públicos y privados gratuitos de Git, solicitudes de extracción y revisiones del código.

- [Azure Pipelines](#)
Puedes usar Azure Pipelines para crear canalizaciones de versión y publicación que automatizan las versiones y las implementaciones.
- [Azure Test Plans](#)
Puedes utilizar Azure Test Plans para mejorar la calidad general de tu código con servicios manuales y de pruebas de exploración para tus aplicaciones.
- [Azure Artifacts](#)
Puedes usar Azure Artifacts para compartir paquetes de código (como paquetes npm, NuGet y Maven) en toda tu organización.

Ahora vamos a explorar los servicios de Azure DevOps de manera más detallada.



Azure Boards

Planificar tu trabajo y supervisar tu progreso son tareas importantes, y Azure Boards puede ayudarte a completarlas.

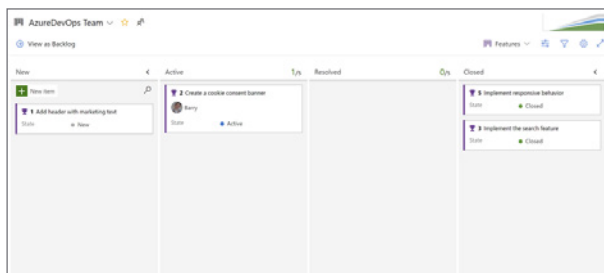


Figura 7-1

En Azure Boards puedes crear un backlog completo de elementos de trabajo (como historias de usuarios) y planificarlos en sprints para que tu equipo pueda trabajar de forma iterativa para terminar las tareas.

Todo el sistema de planificación está optimizado para trabajar de forma ágil. Incluye incluso paneles Kanban para gestionar tu trabajo (Figura 7-1).

Todo se puede personalizar para que funcione lo mejor posible para tus equipos, ya sea mediante el proceso de "scrum", otro método ágil o el de Integración del Modelo de Madurez de las Capacidades (CMMI). Puedes crear y gestionar tareas, funciones, historias de usuarios, errores, requisitos, problemas, solicitudes de cambios y mucho más.

Intenta personalizar tus paneles y crear gráficos (como gráficos de evolución "burndown" o listas de tareas) que muestran la información que necesites. Puedes consultar elementos de trabajo y su progreso y utilizar esta información para personalizar tus paneles, gráficos y listas. A partir de ahí, puedes compartirlas o anclarlas en tu panel de Azure DevOps para que todo el mundo lo vea.

// Pruébalo: Empieza a utilizar Azure Boards para realizar un seguimiento de los problemas, tareas y acontecimientos

Azure Repos

El control de versiones es esencial para trabajar en conjunto y garantizar que tu activo más importante (el código) se almacene de forma segura. [Azure Repos](#) es un conjunto de herramientas de control de versiones para almacenar tu código y compartirlo con tu equipo. Esto resulta útil tanto para los equipos como para los desarrolladores individuales. El control de versiones mantiene un historial de tu desarrollo para que puedas revisar o incluso recuperar cualquier versión anterior del código.

Puedes elegir entre los siguientes dos sistemas de control de versiones al usar Azure Repos:

Git

Este es un sistema de control de versiones ampliamente utilizado entre los desarrolladores, y también es la base de [GitHub](#). [Git](#) es un sistema de control de versiones distribuidas, por lo que todo el código fuente (todas las versiones de todos los archivos) se encuentra en tu equipo, lo que simplifica mucho el trabajo sin conexión. Con Git, la fuente de autoridad está esencialmente en el equipo de todos los usuarios, y se sincroniza cuando los desarrolladores envían su código al servidor Git (en este caso, Azure Repos).

Azure Repos utiliza Git estándar. Esto implica que puedes usarlo con cualquier herramienta o IDE de Git, incluidas [Visual Studio](#) y [Visual Studio Code](#), además de Git para [Windows](#), [Mac](#), [Eclipse](#) e [IntelliJ](#).

Al seguir el [flujo de trabajo de Git](#), normalmente se empieza por crear tu propia rama del código para, por ejemplo, añadir una función. Al terminar, se confirma el código para crear una solicitud de extracción (pull) para esa rama y enviarla al servidor. Los usuarios pueden ver, revisar, probar y debatir sobre esta solicitud pull. Cuando el código sea lo suficientemente bueno como para insertarlo en la rama principal, la solicitud se aceptará y se podrá eliminar la rama de desarrollo.

Con Azure Repos, tendrás un conjunto de herramientas rico sobre el que basar el [flujo de trabajo de Git](#). Puedes vincular elementos de trabajo, como historias de usuarios o errores, a las solicitudes pull, de modo que sepas de qué va cada cambio. Puedes cuestionar el código confirmado e incluso comentar los cambios en él. Azure Repos también permite votar sobre los cambios en el código, de modo que solo se acepta un cambio cuando todos los miembros del equipo afirman que están de acuerdo.

Azure Repos ofrece repositorios Git privados gratuitos e ilimitados.

```
// Empieza aprendiendo a escribir  
código con Git
```

Control de versiones de Team Foundation

El control de versiones de Team Foundation (TFVC) es un sistema de control de versiones centralizado que garantiza que se mantenga una única fuente de autoridad siempre en el servidor. Normalmente, los desarrolladores solo tienen una versión de cada archivo en su equipo, lo que hace que sea más difícil trabajar sin conexión.

Con [TFVC](#), puedes decidir trabajar en los siguientes espacios de trabajo:

Espacios de trabajo en servidores: los desarrolladores descargan de forma pública archivos del servidor para que solo ellos puedan realizar cambios en los archivos. Tras hacer los cambios, pueden volver a subir los cambios y otros desarrolladores pueden descargar el archivo para realizar sus cambios. Así, se elimina la necesidad de fusionar los cambios y elimina la posibilidad de que se den conflictos entre el código.

Espacios de trabajo locales: con ellos, los desarrolladores tienen la última versión de los archivos en sus equipos y pueden cambiarlos. Tras haber terminado de realizar los cambios, pueden subirlos al servidor y resolver los conflictos según sea necesario.

Con TFVC en Azure Repos, todo el mundo puede descargar las versiones de las ramas de código que crees en el servidor. Azure Repos también facilita un rico conjunto de herramientas que te permite adjuntar elementos de trabajo a los cambios en el código. También es posible solicitar y realizar revisiones del código, por lo que tu equipo podrá debatir sobre los cambios y recomendar actualizaciones antes de que se fusionen en la rama principal.

```
// Pruébalo: Empieza a desarrollar  
y compartir tu código en TFVC  
con Visual Studio
```

Azure Pipelines

Cuando tu código esté ya en un repositorio como Azure Repos, puedes empezar a automatizar tu versión y tus procesos de publicación con [Azure Pipelines](#).

Azure Pipelines facilita mucho valor en una cantidad breve de tiempo. Permite la integración continua (CI) para la compilación y la comprobación del código cuando llegan los cambios, además del desarrollo continuo (CD) para la implementación de aplicaciones después de compilar y probar los cambios correctamente. Impulsamos a todas las organizaciones a que exploren el CI y el CD, ya que estos procesos mejoran la calidad del código y reducen los esfuerzos de implementación.

Azure Pipelines puede ayudarte con el CI y el CD al ofrecer canalizaciones de desarrollo e implementación. Cada una de ellas contiene pasos para compilar y comprobar el código e implementarlo en uno o varios entornos. El punto fuerte de Azure Pipelines es que funciona con cualquier tipo de código, independientemente de dónde lo almacenes: desde C# en Azure Repos hasta Java en BitBucket, entre muchos otros.

Azure Pipelines funciona muy bien con los servicios de Azure para implementar tu aplicación en una aplicación web de Azure, por ejemplo. También funciona con cualquier servicio que se ejecute en cualquier otro entorno, como Google Cloud,

Amazon, o incluso on-premises, en tu propio centro de datos. Si ya utilizas herramientas de integración continua como [Jenkins](#) o [Spinnaker](#), puedes traer tus desarrollos y procesos a Azure y aprovechar los complementos de agente dinámico para reducir los requisitos y costes de infraestructura.

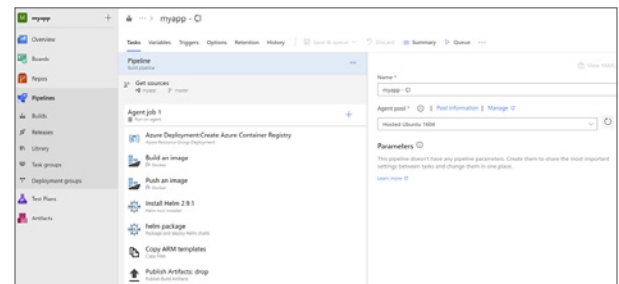


Figura 7-2: Azure Pipelines: canalización de compilación

Hay dos formas de trabajar con Azure Pipelines. Puedes [crear canalizaciones](#) con el diseñador visual en el portal de Azure DevOps o puedes usar el enfoque basado en YAML, más avanzado. En este enfoque, crearás un archivo de código YAML, que contiene todos los pasos de la canalización, y lo confirmarás en el control de la fuente.

La forma más fácil de empezar es usar el diseñador visual. Veamos un ejemplo de una canalización de compilación y versión:

Canalización de compilación: la canalización de compilación (o canalización CI) de la figura 7-2 muestra una lista de tareas que se realizarán cuando se ejecute esta canalización. La canalización está configurada para ejecutarse en cuanto se confirmen

nuevos cambios en Azure Repos. Tomará el código de Azure Repos (que es una aplicación Node.js), creará una imagen de contenedor Docker a partir de él y lo enviará a [Azure Container Registry](#). A partir de ahí, [Helm](#) podrá usarlo para compilar la imagen en un paquete que se podrá implementar en [Azure Kubernetes Service](#).

Ten en cuenta que puedes configurar qué hospedaje ejecuta la canalización por ti. Al lado derecho de la imagen puedes ver que esta canalización en particular se ejecutará en un grupo hospedado de máquinas Ubuntu. También hay disponibles hosts Linux y Windows, e incluido en un MacOS hospedado que podrás usar para crear tus aplicaciones para iOS. Este es solo un ejemplo de una canalización de compilación. Se puede crear una para cualquier aplicación imaginable. También puedes integrar pruebas, incluidas pruebas de unidades y de código estático, en la canalización de compilación.

Canalización de versión: la canalización de versión (o canalización CD) se ejecuta en cuanto la canalización de compilación se ejecuta correctamente, aunque también puedes configurarla para que se active manualmente. La canalización de versión de la figura 7-3 contiene nueve tareas que primero crean un clúster de Azure Kubernetes Service y luego implementan el paquete Helm producido en la canalización de compilación en el clúster.

En la figura 7-3 se muestran los pasos de la etapa de desarrollo. Las etapas son como entornos. Puedes configurar una etapa para tus entornos de desarrollo, prueba y producción, etc. También puedes configurar cosas como aprobaciones previas a la implementación, que requieren que alguien

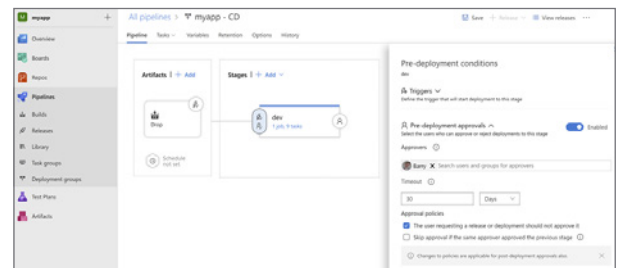


Figura 7-3

apruebe la publicación de una aplicación en un entorno específico (como el de producción). Esto supone que puedes automatizarlo todo y dejar la decisión de publicar en producción a un supervisor, sobre la base de los resultados de las pruebas para los pasos anteriores en la canalización.

Puedes hacer que tu canalización sea lo sencilla o compleja que quieras. Idealmente, conviene automatizar todo lo posible, desde la creación y destrucción de tu infraestructura hasta la implementación y las pruebas de tu aplicación. Las tareas de la canalización están disponibles para casi todo, y puedes acceder a más tareas como extensiones a Azure DevOps en el [Visual Studio Marketplace](#).

// Empieza a usar Azure Pipelines
creando tu primera canalización

Azure Test Plans

Para mejorar la calidad de tus aplicaciones, utiliza [Azure Test Plans](#) para definir planes de prueba y crear y ejecutar pruebas manuales y de exploración. Azure Test Plans facilita las herramientas para crear pruebas, ejecutarlas, registrar el feedback y seguir los resultados de la prueba.

En Azure Test Plans, puedes empezar por crear un plan de prueba. Contiene varios conjuntos y casos de prueba. Un caso de prueba puede ser una prueba de exploración en la que se analiza la aplicación para ver si funciona de la forma esperada; una prueba manual guiada, en la que se describen de forma detallada los pasos de prueba y los resultados esperados; o una prueba automatizada. Puedes incluso grabar los pasos de prueba registrando los clics en una aplicación y dejando que Azure Test Plans automatice los clics en forma de prueba. También puedes incorporar pruebas de estrés y carga en tus canalizaciones de versión y compilación. Los casos de prueba son elementos de trabajo, como las tareas y las historias de usuarios, y se pueden programar en una iteración.

Tras haber creado una prueba (Figura 7-4), puede ejecutarla un evaluador. En una prueba manual, por ejemplo, el evaluador usa la herramienta de prueba para realizar los pasos de la prueba y registrar los hallazgos, incluida la pantalla, la voz del evaluador, las capturas de pantalla y los adjuntos. El evaluador aprueba o suspende cada uno de los pasos de la prueba.

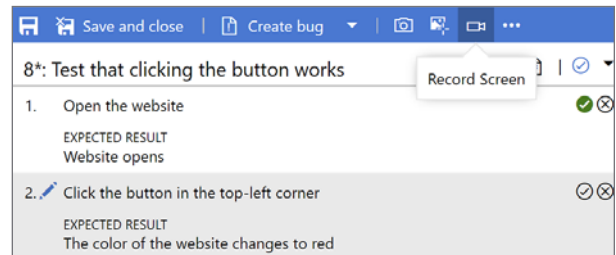


Figura 7-4

A partir del contexto de la prueba, el evaluador puede crear también un error que haya que resolver.

Para garantizar que las expectativas de los participantes estén alineadas con tu plan, Azure Test Plans también te permiten [pedir comentarios](#) sobre elementos de trabajo, como historias de usuarios. Esto permite que los interesados echen un vistazo a lo que propones y te faciliten sus comentarios en forma de texto, adjuntos, vídeo o voz.

Y quizás lo que sea más importante es que Azure Test Plans facilita paneles y gráficos sobre el progreso y el estado de las pruebas en tu proyecto. Puedes usarlos para ver la calidad de tu aplicación y su progreso a lo largo del tiempo. Esto te puede ayudar a identificar funciones que no están listas para implementarse.

```
// Empieza a usar Azure Test  
Plans creando casos de prueba  
manuales
```


Azure Artifacts

Como los paquetes suelen ofrecer una funcionalidad que no tienes que crear tú mismo, seguramente usas muchos en tus aplicaciones. Y es probable que accedas a ellos desde muchas fuentes: NuGet, npm, Maven y muchas otras. ¿Pero qué pasa si tu equipo crea paquetes que quieras usar solo internamente? ¿Dónde las hospedas de forma segura y cómo las compartes? [Azure Artifacts](#) proporciona esta capacidad. Azure Artifacts es una fuente de paquete que te permite hospedar los paquetes que crees y protegerlos para tu organización.

Puedes hospedar [todo tipo de paquetes](#) en Azure Artifacts, incluidos NuGet, npm, Maven, Python y Universal Packages. Puedes utilizar incluso el feed de Azure Artifacts para almacenar paquetes de fuentes públicas, como nuget.org y npmjs.com. Al almacenar paquetes de fuentes públicas en tu feed, podrás seguir usándolos aunque no estén ya disponibles en el feed público. Esto es especialmente útil para paquetes críticos.

Sigue estos sencillos pasos para usar Azure Artifacts:

1. [Crear](#) un feed de Azure Artifacts.
2. [Publicar](#) tu paquete en el feed.
3. [Consumir](#) el feed en tu IDE favorito, como Visual Studio.

```
// Pruébalo: Empieza a trabajar  
con paquetes de Python en Azure  
Artifacts
```

Lectura adicional

Si quieres mejorar la calidad de tu software y aprender más sobre la automatización de tus procesos de compilación y versión, descarga y lee estos e-books gratuitos:

```
// DevOps efectivas
```

```
// Entrega continua en Java
```

```
// Azure para arquitectos
```

08 /

Azure en acción

Tutorial 1: La experiencia del portal de Azure

Una de las herramientas más importantes de Azure es el nodo central: el portal de Azure. La mayoría de las cosas que se pueden hacer en el portal de Azure se pueden hacer también a través de la API de Azure, la CLI de Azure y Azure PowerShell.

El portal de Azure es un panel con mosaicos. Es fácil crear y personalizar los paneles y compartirlos con los miembros del equipo.

Mosaicos en el portal de Azure

Los mosaicos, que aparecen en la figura 8-1, muestran información sobre un servicio o actúan como acceso directo a un servicio. Aparecen en el portal, en las páginas de todos los servicios. Pueden ser una forma muy útil de obtener información general sobre el funcionamiento de un servicio con rapidez.

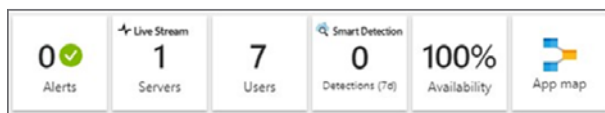


Figura 8-1

El mosaico del servicio Application Insights muestra información, como alertas activas, recepción de datos en directo, usuarios activos en las últimas 24 horas y disponibilidad. También puedes personalizar el tamaño y la información de los mosaicos, además del 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 directamente en los paneles para que sean lo primero que se ve al entrar en el portal (figura 8-2). Por ejemplo, puedes anclar mosaicos de todas las métricas de servicio que utilizas para crear un [panel de supervisión](#) con el fin de compartirlo con el equipo o de visualizarlo en un monitor físico.

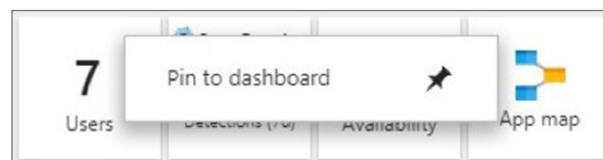


Figura 8-2

Añadir servicios

Puedes encontrar y añadir servicios en el portal de Azure de diversas 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 aplicaciones web hasta servidores Linux, como se muestra en la Figura 8-3.

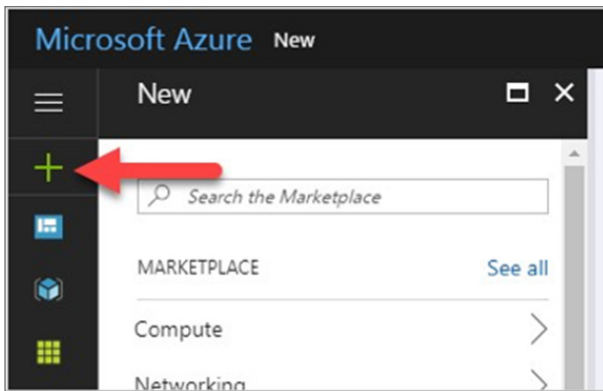


Figura 8-3

Al encontrar el servicio que deseas entre los resultados de búsqueda, como se muestra en la figura 8-4, un asistente te guiará en su configuración e implementación.

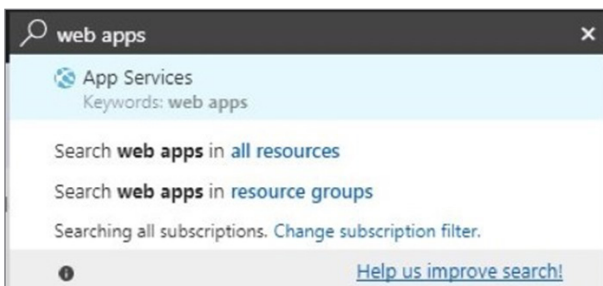


Figura 8-4

Puedes usar el cuadro de búsqueda de la parte superior del portal para buscar en todos tus recursos e ir directamente a ellos (figura 8-5). El menú de favoritos está en el panel al lado izquierdo del portal.

Este menú muestra las categorías de recursos, como Azure App Service, representadas por sus iconos. Puedes reorganizar los iconos arrastrándolos arriba y abajo. También puedes seleccionar los que desees ver expandiendo el menú Favoritos y haciendo clic en el símbolo de estrella junto a esas categorías.

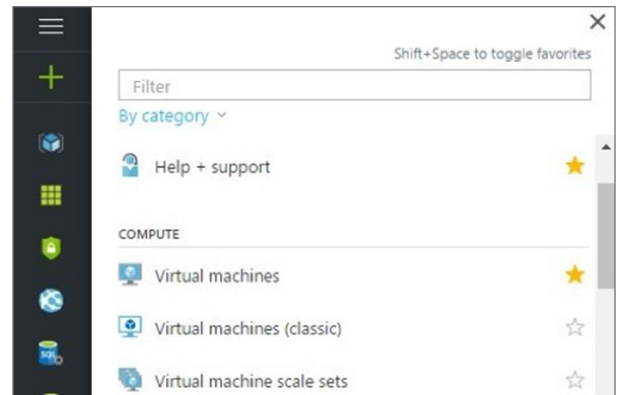


Figura 8-5

Descripción de las hojas

La páginas de Azure también se llaman hojas, y puedes anclarlas a tus paneles. Cuando abres una aplicación web, primero ves la hoja de información general, como en la Figura 8-6.

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. Por ejemplo, si abres la hoja Ranuras de implementación y haces clic en 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.

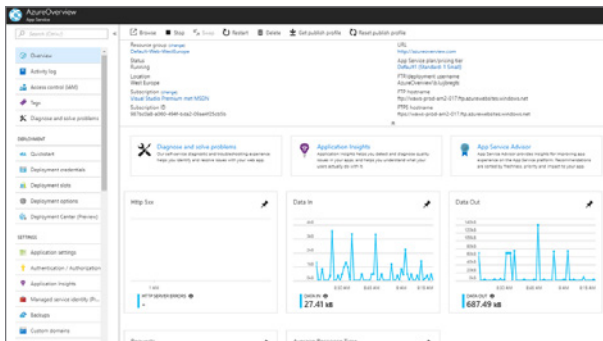


Figura 8-6

Crear una nueva VM

Vamos a usar el portal de Azure para crear una nueva máquina virtual. Tras hacerlo, la apagaremos y eliminaremos para que no sigas pagando por ella.

Una nota sobre los grupos de recursos

La VM se implementará en un grupo de recursos, un contenedor lógico que alberga los recursos. Todos los recursos de Azure residen en grupos de recursos. También 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.

1. En el portal de Azure, en la parte superior izquierda, selecciona **Crear un recurso**.
2. En el cuadro de búsqueda, escribe **Windows Server virtual machine**.
3. Haz clic en **Windows Server 2016 Datacenter**.
4. Haz clic en **Crear**. Se abre el asistente para crear máquinas virtuales.
5. Elige un nombre para la máquina virtual.
6. Elige el tipo de unidad. SSD proporciona una máquina virtual más rápida, pero es más caro. Para este tutorial, elige **SSD**.
7. Escribe un nombre de usuario.
8. Selecciona **Contraseña** en tipo de autenticación.
9. Escribe una contraseña y confírmala.
10. En el cuadro **Grupo de recursos**, escribe un nuevo nombre.

11. Elige la ubicación de la máquina virtual y haz clic en **Aceptar**.
12. Elige el tamaño de la máquina virtual. Hay muchas opciones de tamaños para VM. El rendimiento de la VM determina el coste. Usa el asistente para seleccionar cuántos núcleos y cuánta memoria quieres, y elegir las opciones en función de eso. Además, hay otras características que tienen opciones de tamaño, como:
 - Tipo de disco duro (SSD o disco duro normal).
 - Cantidad máxima de IOPS (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.
 - Cantidad de unidades de datos que se pueden instalar en la máquina virtual.
 - Capacidad de equilibrar la carga.
 - Tarjeta gráfica instalada en la máquina virtual. Esto resulta útil si necesitas realizar un renderizado de gráficos sustancial o si hay cargas de trabajo computacionales altas.
13. Después de seleccionar el tamaño, puedes configurar ajustes adicionales como la red virtual, la dirección IP y las extensiones en la máquina. Por ahora, déjalo todo como está y selecciona **Aceptar**.

14. Repasa el resumen, acepta los términos y haz clic en **Crear**.

La implementación de la VM suele tardar solo unos minutos. Cuando navegas a la máquina virtual en el portal de Azure, puedes seguir configurándola e iniciar sesión con el protocolo de escritorio remoto (RDP).

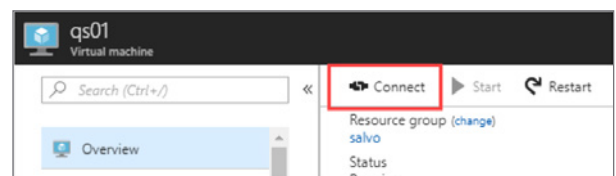


Figura 8-7

Para iniciar sesión en la máquina virtual con RDP, haz clic en el botón **Conectar** en la hoja de **información general** de la máquina virtual en el portal de Azure (Figura 8-7). Esto activa una descarga del archivo RDP, que puedes utilizar para conectarte a la máquina virtual.

Limpieza de los recursos del tutorial

Cuando hayas terminado de usar la máquina virtual, puedes cerrarla y quitarla eliminando el grupo de recursos que creamos al generar la máquina virtual. Este grupo contiene la máquina virtual y todos los demás recursos que se crean automáticamente. Tras eliminar el grupo de recursos, ya no pagarás por ninguno de los recursos que utilizaste en este tutorial.

Tutorial 2: Desarrollar una aplicación web y una base de datos en Azure

En este tutorial, vamos a implementar una aplicación sencilla de .NET Core que se conecte con una base de datos SQL. Después, las hospedaremos en Web Apps.

Para continuar, necesitarás [Git v2 o superior](#), [.NET Core](#) y [Visual Studio Code](#) instalados en tu dispositivo. También utilizaremos una muestra de la aplicación ASP.NET Core MVC para administrar la lista de pendientes.

Crear la aplicación web y la base de datos con el portal de Azure

Para hospedar la aplicación de .NET Core, vamos a crear una nueva aplicación web en el portal de Azure.

1. En el portal de Azure, selecciona **Crear un nuevo servicio**.
2. Busca **Aplicación web**. Se abre la hoja de **Aplicación web**. Selecciona **Crear**. Se abre la hoja de **Crear aplicación web**.

3. Escribe un nombre para la aplicación web.
4. Crea un nuevo grupo de recursos asignándole un nombre.
5. Deja la selección de sistema operativo en **Windows**.
6. Selecciona o crea un **plan de App Service** y haz clic en **Crear**.

App Services, al igual que Web Apps, se ejecuta con planes de Azure App Service. Los planes de App Service son una abstracción de recursos y características, como la CPU y la memoria, y se representan por medio de niveles de precios.

Los planes de App Service también están vinculados a la región geográfica que elijas. Puedes, por ejemplo, ejecutar la aplicación de Web Apps en un plan de App Service del nivel de precios S1, que tiene 1 núcleo y 1,75 GB de RAM, como se muestra en la figura 8-8.



Figura 8-8

Puedes ejecutar tantos App Services como quieras en un plan de App Service, pero ten en cuenta tienes que compartir los recursos entre todos los App Services.

Para hospedar la base de datos, crearemos un base de datos SQL. Funciona de igual manera que una base de datos de SQL Server local y ahora se ejecuta completamente administrada en Azure.

1. En el portal de Azure, haz clic en **Crear un nuevo servicio**.
2. Busca **SQL Database** y haz clic en la hoja **SQL Database**. Haz clic en **Crear**. Se abre la hoja de **creación de la base de datos SQL**.
3. Escribe un nombre para la base de datos.
4. Selecciona el grupo de recursos que creaste para la aplicación web.
5. Deja la fuente como **Base de datos en blanco**.
6. Haz clic en **Servidor** para crear un nuevo servidor de base de datos SQL.
 - a. Escribe un nombre para el servidor.
 - b. Escribe el inicio de sesión de administrador del servidor. Este es el nombre de usuario del servidor.
 - c. Escribe la contraseña que utilizarás para iniciar sesión en el servidor.
 - d. Confirma la contraseña.
 - e. Elige una ubicación. Elige la misma ubicación que seleccionaste para el plan de App Service.
 - f. Haz clic en **Seleccionar** para enviar la configuración del nuevo servidor.
7. Selecciona un nivel de precios. Para fines de pruebas y desarrollo, el nivel **Básico** es suficiente.
8. Haz clic en **Crear**. Ahora se creará la base de datos.
9. Desplázate hasta la base de datos SQL y haz clic en **Mostrar cadena de conexión a la base de datos**.
10. Toma nota de la cadena de conexión, porque la necesitarás después en este tutorial.

Ejecutar la aplicación .NET Core localmente

Vamos a ejecutar la app localmente antes de ejecutarla en Azure. La aplicación puede ejecutarse localmente debido a que, de forma predeterminada, utiliza una base de datos SQLite, la cual es un motor de base de datos SQL autónomo.

1. Abre una línea de comando y dirígete al directorio que quieras utilizar como directorio para el código fuente en este proyecto.
2. Ejecuta los siguientes comandos para obtener el código fuente y navega hasta la carpeta del proyecto:

```
git clone https://github.com/azure-samples/dotnetcore-sqldb-tutorial  
  
cd dotnetcore-sqldb-tutorial
```

3. El proyecto utiliza Entity Framework Core para rellenar su base de datos. Para asegurar que la base de datos esté actualizada y ejecutar la aplicación de forma local, ejecuta los siguientes comandos:

```
dotnet restore  
  
dotnet ef database update  
  
dotnet run
```


4. La aplicación debería estar ahora en ejecución y la URL a la app (como **http://localhost:5000**) debe encontrarse en la salida de la ventana de comandos.
5. Dirígete a esa URL en un navegador. Esto cargará la aplicación y aparecerá como en la Figura 8-9. A continuación, puedes crear nuevos elementos haciendo clic en el enlace **Crear nuevo**.
6. Puedes cerrar la aplicación si cierras la ventana de comandos o al presionar **Ctrl+C**.

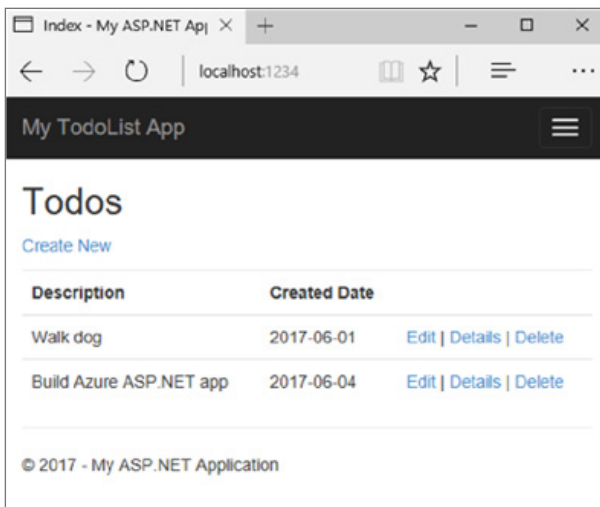


Figura 8-9

Conectar la aplicación de web local a la base de datos que se ejecuta en Azure

Ahora tienes una aplicación funcional que se ejecuta localmente. Antes de implementarla en Azure, cambiaremos el código fuente para que pueda conectarse a la base de datos SQL.

1. En tu repositorio de código fuente, encuentra el archivo **Startup.cs** y localiza el siguiente código:

```
services.AddDbContext<MyDatabaseContext>
(options => options.UseSqlite
("Data Source=localdatabase.db"));
```

2. Reemplaza el código con el siguiente código, el cual se puede conectar con la base de datos de Azure SQL:

```
// Use SQL Database if in Azure, otherwise,
use SQLite

if(Environment.
GetEnvironmentVariable("ASPNETCORE_
ENVIRONMENT") == "Production")

services.
AddDbContext<MyDatabaseContext>(options =>

options.UseSqlServer(Configuration.
GetConnectionString("MyDbConnection"));
else

services.
AddDbContext<MyDatabaseContext>(options =>

options.UseSqlite("Data
Source=localdatabase.db"));

// Automatically perform database migration
services.BuildServiceProvider().
GetService<MyDatabaseContext>().Database.
Migrate();
```

Este código observa el entorno en que se ejecuta y cambia su conexión de base de datos en función de esa información. Cuando se ejecuta en el entorno de producción (que en nuestro caso es Azure), el código obtendrá la cadena de conexión para la base de datos desde la variable `MyDbConnection`, que configuraremos en Azure.

El código ejecuta también el método `Database.Migrate`, que ejecuta las migraciones de Entity Framework Core que ejecutamos anteriormente de forma manual.

3. Guarda los cambios y ejecuta los siguientes comandos para enviar los cambios al repositorio Git local:

```
git add .  
git commit -m "connect to SQLDB in Azure"
```

Ahora configuraremos la variable de la cadena de conexión en Azure.

4. En el portal de Azure, dirígete a la aplicación web que creamos anteriormente.
5. Desplázate hasta **Configuración de la aplicación**.
6. Crea una nueva cadena de conexión denominada **MyDbConnection**. El valor debe ser la cadena de conexión a la base de datos SQL (incluidos el nombre de usuario y la contraseña) que guardaste antes, al crear la base de datos.

7. Haz clic en **Guardar**. La configuración de la aplicación en el portal de Azure debería aparecer como en la Figura 8-10.

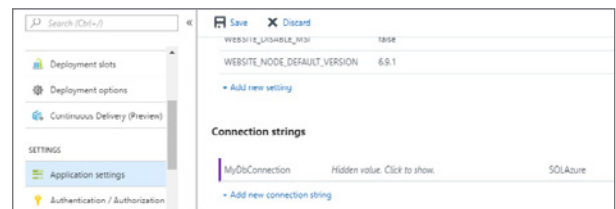


Figura 8-10

Implementación de la aplicación web en Azure

Vamos a utilizar la Git para transferir la aplicación a Azure. Para conectar el repositorio Git local con Azure, debes tener configurado un usuario de implementación en el servidor (Azure Web App) 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.

1. En el portal de Azure, navega a **Azure Cloud Shell** seleccionando el botón en la barra superior que tiene este aspecto:



2. Azure Cloud Shell te permite utilizar la CLI de Azure en el cloud y se encarga de la autenticación. Cuando Cloud Shell está completamente cargada, ejecuta el siguiente comando para crear al usuario de implementación. Cambia los valores **<usuario>** y **<contraseña>** por los que hayas creado. Toma nota del nombre de usuario y la contraseña, porque los necesitarás después.

```
az webapp deployment user set --user-name  
<username> --password <password>
```

3. El comando dará como resultado una salida JSON. Si aparece un error **'Conflict'. Details: 409**, cambia el nombre de usuario. Si aparece un error **'Bad Request'. Details: 400**, debes crear una contraseña más segura.

Ahora enviaremos el código fuente desde nuestro repositorio local Git hasta la aplicación web de Azure.

4. Abre la línea de comandos en tu equipo local.
5. Añade un Azure remoto a tu repositorio Git local utilizando la URL de Git remoto:
 - a. Cambia el **<usuario>** con el nombre de usuario utilizado al crear al usuario de implementación.
 - b. Cambia el **<nombre_de_la_app>** con el nombre de la aplicación web de Azure.
 - c. Utiliza la URL para ejecutar el siguiente comando:

```
git remote add azure <deploymentLocalGitUrl>
```

6. Cuando se haya añadido el destino remoto al repositorio Git, puedes insertar tu código al ejecutar el siguiente comando. Tendrás que introducir las credenciales para insertar el código a Azure. Utiliza el nombre de usuario y la contraseña empleados para crear el usuario de implementación.

```
git push azure master
```

La primera vez que insertas el código fuente a Azure podría tardar unos minutos. Cuando se haya terminado el proceso, dirígete a la URL de tu aplicación web de Azure, que tendrá este aspecto: **http://<app_name>.azurewebsites.net**

7. Agrega algunos elementos pendientes en la app para probar la conexión a la base de datos.

Ahora tienes una aplicación funcional que se ejecuta en Azure.

Tutorial 3: Expandir las aplicaciones con Logic Apps y Cognitive Services

Una característica de gran alcance de nuestra aplicación es la capacidad de analizar el contenido de las tareas pendientes y luego crear automáticamente citas en el calendario para las tareas que requieren una fecha específica.

Por ejemplo, si un usuario crea un nuevo elemento de tareas pendientes con el texto "cena familiar siguiente viernes 19:00", la aplicación creará un elemento de calendario para ese viernes específico a las 19:00 con el tema "comida familiar".

Configuraremos esta característica utilizando la función [Logic Apps](#) de Microsoft Azure App Service y el [Language Understanding Intelligent Service \(LUIS\)](#), de la siguiente forma:

- La aplicación .NET Core escribe la lista de tareas pendientes en la base de datos SQL.
- La aplicación lógica se activa por cada nueva fila que se crea en la base de datos.
- La aplicación lógica toma el texto de la lista de tareas pendientes y lo pasa al servicio Language Understanding.
- El servicio Language Understanding analiza el texto y crea un elemento de calendario en tu calendario de Office 365, si el texto contiene fecha y hora.

No tenemos que cambiar nuestra aplicación de ninguna forma para añadir esta funcionalidad. Logic Apps y Cognitive Services son servicios adicionales que simplemente analizan los datos que ya existen.

Manos a la obra.

Crear un servicio de Language Understanding

Primero crearemos el servicio de Language Understanding para poder usarlo después en nuestra aplicación lógica. En este ejemplo, mantendremos el modelo simple y no desarrollaremos nada sobre él para que esté listo para cada variación que los usuarios puedan necesitar para una fecha en el elemento pendiente. Puedes añadir tú un modelo propio en lugar de usar el que hemos creado.

1. En el portal de Azure, selecciona **Crear un nuevo servicio**.
2. Busca **Language Understanding** y selecciónalo en los resultados de búsqueda para abrir la hoja **Language Understanding**. Selecciona **Crear**. Se abre la hoja para **Crear Language Understanding**.
3. Escribe un nombre.
4. Selecciona un nivel de precios (cualquiera funciona para este tutorial).
5. Crea un nuevo grupo de recursos llamado **datadetection**.
6. Haz clic en **Crear**.
7. Navega al servicio Language Understanding cuando se haya creado.
8. De forma predeterminada, el servicio se abrirá en la hoja de **Inicio rápido**. A partir de aquí, selecciona **Portal de Language Understanding**.
9. De ser necesario, inicia sesión en **Iniciar sesión** en la esquina superior derecha.

10. Haz clic en **Crear nueva app**.

11. Escribe un nombre.

12. Haz clic en **Listo**.

Ya estamos en el portal de Language Understanding y podremos crear un modelo de lenguaje. Queremos que el servicio de Language Understanding pueda comprender la frase "cena familiar siguiente viernes 19:00." Para ello, primero agregaremos algunas entidades, que son elementos del texto que el servicio reconocerá.

1. Haz clic en **Entidades**.
2. Haz clic en **Administrar entidades predesarrolladas**.
3. Selecciona **Datetimev2** y **keyPhrase**.
4. Haz clic en **Listo**. Ahora tenemos dos entidades que reconocerán texto por nosotros.
5. Haz clic en **Intenciones**.
6. Haz clic en **Crear nueva intención**.
7. Escribe un nombre, como "Agregar pendiente al calendario", que es la intención que queremos detectar en el texto.
8. Haz clic en **Listo**.

Ahora puedes escribir **expresiones**. Son textos de muestra que representan la intención que queremos detectar.

9. Introduce "cena familiar siguiente viernes 19:00" para representar la intención de agregar un elemento pendiente al calendario.
10. Como ya hemos agregado dos entidades, el texto en la expresión se analiza y se reconoce como tales entidades, como se muestra en la figura 8-11. El texto "cena familiar" se reconoce como una **keyPhrase**. El texto "siguiente viernes 19:00" se reconoce como **datetimev2**.
11. Utilicemos este modelo para entrenar al servicio y publicarlo. Selecciona **Entrenar** en la esquina superior derecha de la pantalla.

keyPhrase	datetimev2
-----------	------------

Figura 8-11

- Esto realiza entrenamiento de machine learning y desarrolla un modelo de machine learning basado en lo que acabamos de introducir.
 - Para probar si el servicio funciona como se esperaba, escribe "cena familiar siguiente viernes 19:00" en la ventana **Prueba** junto al botón **Entrenar**.
12. Ahora que tenemos un servicio funcional, necesitamos publicar el modelo para producción. Haz clic en **Publicar** en el menú (junto al botón **Entrenar**) para que aparezca la página **Publicar**.
 13. Deja la ranura como **Producción**.
 14. Haz clic en **Publicar**.

15. El modelo se publicará a producción. Baja hasta **Recursos y claves** y toma nota de la cadena clave que encontrarás ahí, porque la necesitaremos para nuestra aplicación lógica.

Crear la aplicación lógica

La aplicación lógica que creamos se activará con las nuevas filas de elementos pendientes escritos en SQL Database. Después tomará el valor de cada elemento pendiente y lo enviará al servicio de Language Understanding para analizarlo. Si el servicio Language Understanding encuentra una fecha en el elemento, creará un nuevo evento de calendario en tu cuenta de Office 365.

Vamos a crear la aplicación lógica:

1. En el portal de Azure, haz clic en **Crear un nuevo servicio**.
2. Busca **aplicación lógica** y haz clic en los resultados de la búsqueda para abrir la hoja **Aplicación lógica**. Haz clic en **Crear**. Se abre la hoja **Crear aplicación lógica**.
3. Escribe un nombre.
4. Selecciona el grupo de recursos que creaste para el servicio de Language Understanding.
5. Elige una ubicación.
6. Haz clic en **Crear**.
7. Tras crear la aplicación lógica, verás una página de inicio rápida que te preguntará si quieres iniciar la aplicación lógica a partir de una plantilla (figura 8-12). Selecciona **Aplicación lógica en blanco**.

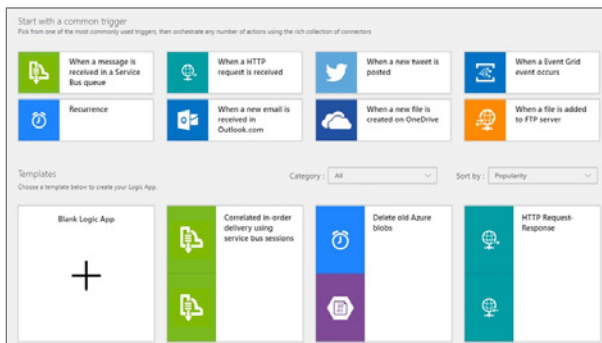


Figura 8-12

Ahora tenemos que crear un disparador para la aplicación lógica.

1. Busca SQL.
2. Selecciona la tarea **Cuando se crea un elemento**. Pedirá la conexión a la base de datos SQL (figura 8-13).

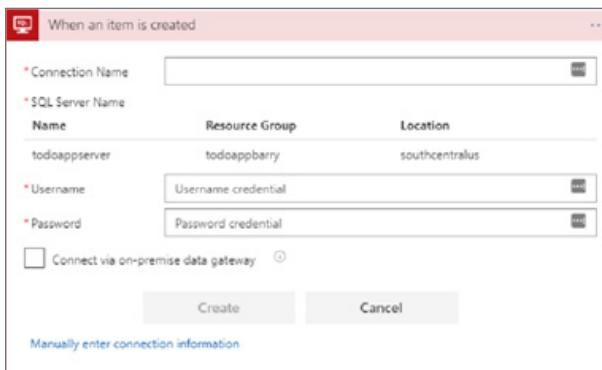


Figura 8-13

3. En este caso, el SQL Server correcto ya está seleccionado, dado que solo hay uno. Quizá debas seleccionar el servidor apropiado.
4. Escribe un nombre para la conexión y después escribe el nombre de usuario y la contraseña para la base de datos SQL que habíamos creado.
5. Haz clic en **Crear**. Esto crea la conexión y la guarda en tu suscripción de Azure. Puedes volver a utilizar esta conexión en otras aplicaciones lógicas.

6. Selecciona la tabla que queremos supervisar, la tabla **Pendientes**.
7. Selecciona un intervalo y una frecuencia. Algunos disparadores de las aplicaciones lógicas se tienen que sondear para dispararse, mientras que a otras se les envía la información por push.

Ahora la aplicación lógica se disparará cada vez que introducimos un nuevo elemento de la lista de pendientes.

A continuación añadiremos otra acción para la aplicación lógica.

1. Haz clic en el **signo de más** en la tarea SQL y después selecciona **Agregar una acción** para agregar la siguiente acción (figura 8-14).

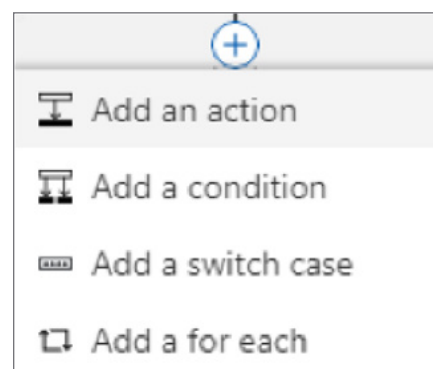


Figura 8-14

2. Busca **LUIS**, lo que abrirá el servicio de Language Understanding. Selecciona la **acción Get prediction de LUIS**. Pedirá una conexión al servicio de Language Understanding.

3. Escribe un nombre para la conexión.
4. Pega la clave de conexión que habías guardado cuando publicamos el modelo de Language Understanding.
5. Haz clic en **Crear**.
6. Selecciona la ID de aplicación que creaste en el portal de Language Understanding.
7. Selecciona la descripción de la tarea SQL como entrada para el campo **Expresión**.
8. Selecciona el elemento **Agregar pendiente al calendario** para la Intención deseada. El resultado comunicará si la tarea contiene una fecha o no.
9. Haz clic en el **signo de más** y después haz clic en **Añadir una condición**. Probaremos si el texto contiene una fecha al confirmar si la intención deseada era verdadera. Si el texto contiene una fecha, crearemos un evento de calendario. Si no, no haremos nada.
10. En la condición, selecciona el valor **Es intención deseada** desde la tarea Language Understanding para el valor.
11. No modifiques la declaración **es igual que**.
12. Añade **verdadero** en el cuadro de texto del valor.
13. La condición aparece tanto en el cuadro **si verdadero** como en el de **si falso**. En el cuadro **si verdadero**, crea una nueva acción.
14. Busca LUIS, como hicimos antes.
15. Selecciona la acción **Obtener entidad por tipo**. Esta es una acción de Language Understanding que extrae una entidad con base en su tipo de los resultados de Language Understanding.
16. Selecciona la ID de aplicación como hicimos anteriormente.
17. Selecciona **builtin.datetimev2** para la entidad deseada.
18. Selecciona el objeto **LUIS Prediction** del campo **luisPredictionObject**.
19. Debajo de esta acción, agrega otra para **Obtener entidad por tipo**.
20. Selecciona la ID de aplicación.
21. Selecciona **builtin.keyPhrase** para la entidad deseada.
22. Selecciona el objeto **LUIS Prediction** del campo **luisPredictionObject**.
23. Crea una nueva acción mientras sigues en el cuadro **si verdadero**. Busca **Office 365** y después selecciona la acción **Crear evento V2**. Esta puede crear un evento en tu calendario de Office 365.
24. Ten en cuenta que esto requiere una conexión a Office 365. Haz clic en **añadir nueva conexión** e inicia sesión con tus credenciales de Office 365. La aplicación lógica mantendrá la conexión en tu suscripción de Azure.
25. Selecciona el calendario para crear un evento.
26. En los campos **Hora de finalización** y **Hora de inicio**, selecciona el **Valor de entidad** de la acción donde filtraste la entidad **datetimev2**.
27. En el campo **Tema**, selecciona el **valor de la entidad** de la acción donde filtraste la entidad **keyPhrase**.
28. Guarda el flujo de la aplicación lógica. El cuadro **si sí** debería tener un aspecto similar al de la figura 8-15.
29. Dirígete a la URL de la aplicación de tareas, que es la URL de tu aplicación web del tutorial anterior.

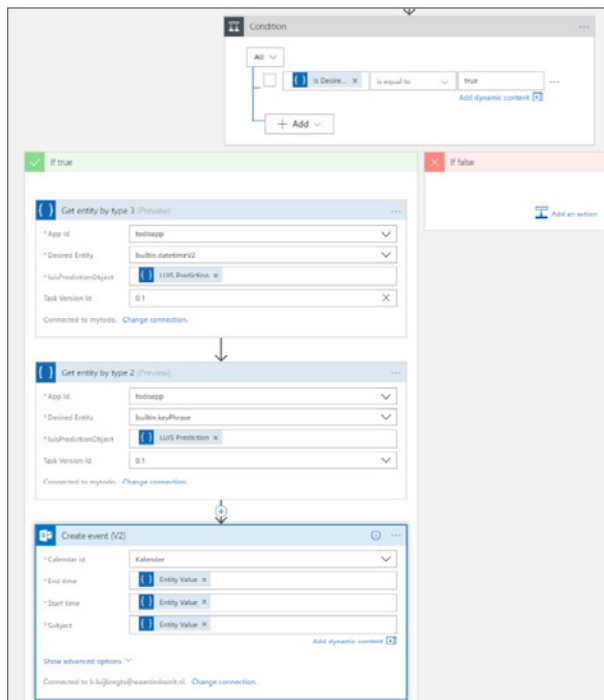


Figura 8-15

30. Crea un nuevo elemento pendiente con el texto “cena familiar siguiente viernes 19:00”. Esto deberá crear un evento en tu calendario.

Además de comprobar tu calendario, puedes ver cómo se ejecutó la aplicación lógica si revisas el **Historial de ejecuciones**. Puedes obtener acceso al historial de ejecuciones al abrir la aplicación lógica desde el portal de Azure. Desde el historial de ejecuciones, puedes incluso reenviar el valor para ejecutarlo nuevamente a través de la aplicación lógica.

Este ejemplo muestra que puedes ampliar una aplicación con los servicios de Azure solo mediante la configuración y sin cambiar el código.

Hicimos este ejemplo simple para que sea fácil seguir este tutorial. En una situación real, el modelo de Language Understanding debe ser más robusto para poder comprender más expresiones. Además, podrías tener el desencadenante de la aplicación lógica en las ediciones a los elementos de tareas, no solo durante su creación.

Tutorial 4: Listo para producción

Con tu aplicación ya en ejecución, ahora puedes utilizar Azure para mejorarla y hacerla más fácil de actualizar.

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 Azure DevOps Repos o GitHub.

Vamos a usar GitHub para transferir nuestro código y después enlazarlo a nuestra aplicación web para que los cambios se implementen automáticamente en una canalización de entrega continua.

Manos a la obra.

1. Inicia sesión desde <https://github.com/new> para crear un nuevo repositorio en GitHub.
2. Escribe un nombre para el repositorio.
3. Deja los otros ajustes como están (repositorio público, no crear un archivo README).
4. Crea el repositorio, que debería tener un aspecto similar al de la figura 8-16.

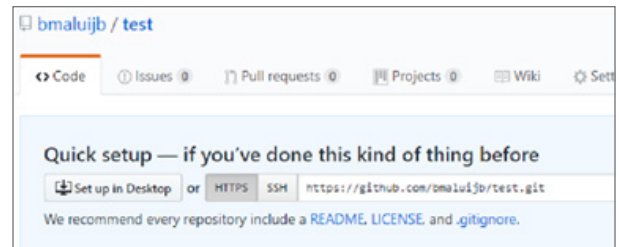


Figura 8-16

5. Usa la dirección URL que aparece en la pantalla para establecer un destino remoto para el repositorio local de Git. Puedes hacerlo en la ventana de comandos.
6. Utiliza el comando de CD para cambiar al directorio del código fuente de la aplicación.
7. Ejecuta el siguiente comando:

```
git remote add github https://github.com/  
bmaluijb/test.git
```

8. Ejecuta el siguiente comando para insertar el código en GitHub:

```
git push github
```

Con eso, el código está en GitHub y se puede compartir con el equipo.

Vamos a configurar la entrega continua con la característica Opciones de implementación de Web Apps a través del portal de Azure. Ten en cuenta que también podemos usar la característica de entrega continua en Web Apps directamente, pero eso requiere una cuenta de Azure DevOps Services.

1. En el portal de Azure, ve a la aplicación web que hospeda la aplicación de tareas pendientes de .NET Core.
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. De ser así, haz 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 la rama configurada como **maestra**.
8. Haz clic en **Aceptar**.
9. Vuelve al menú **Opciones de implementación**. Ahora puedes ver que GitHub 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. Esto se muestra en la figura 8-17, que ilustra la hoja de **Opciones de implementación**. También puedes forzar este proceso haciendo clic en **Sincronizar**.

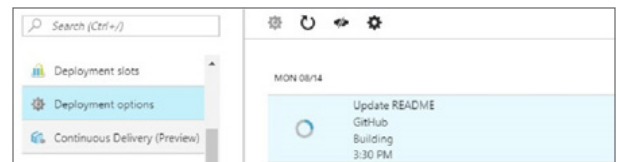


Figura 8-17

Configuración de entornos de ensayo

Con las aplicaciones web de Azure App Service, puedes configurar una ranura de ensayo para probar nuevas versiones de tu aplicación mediante ranuras de implementación. Las ranuras de implementación son App Services completos con los que puedes probar el código antes de promoverlo a la siguiente ranura.

Hay 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 .NET Core. 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. Recuerda 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. Debes ejecutar 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 después intercambiarla con la versión en la ranura de producción. Esto prepara la aplicación antes del intercambio, lo que se traduce en una implementación sin tiempo de inactividad.

Veamos cómo crear una ranura de implementación y cómo intercambiarla.

1. En el portal de Azure, ve a la aplicación web que hospeda la aplicación de .NET Core.
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.
4. Escribe un nombre para la ranura, por ejemplo "ensayo".
5. Elige la **aplicación web de .NET Core** como el origen de configuración. Esto copia la configuración de la aplicación en la nueva ranura.
6. Haz clic en **Aceptar** para crear la ranura, lo cual es similar a la aplicación web original.
7. Configura CD para la ranura como lo hiciste en la aplicación web.
8. Desconecta la conexión de CD en la aplicación web original de .NET Core. Así, cuando se inserta nuevo código, se entrega solamente en la ranura de ensayo.
9. En la aplicación .NET Core, cambia parte del texto en el archivo **Index.cshtml** de la carpeta **Vistas/Inicio**.
10. Confírmalo en Git e insértalo en GitHub, tal como al implementar la aplicación de .NET Core.

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 .NET Core 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.

Ahora vamos a poner la nueva versión en producción.

1. En el portal de Azure, ve a la aplicación web de .NET Core.
2. En la barra de menús, selecciona **Ranuras de implementación** para abrir la hoja **Ranuras de implementación**.
3. Haz clic en el botón **Intercambiar** para abrir la hoja **Intercambiar**. Deja todos los ajustes como están.
4. Haz clic en **Aceptar** para iniciar el intercambio.

Tras finalizar el intercambio, la nueva versión de la aplicación web de .NET Core se encontrará en producción. Puedes probarla si te diriges a la URL de la aplicación web de Node.js. Usar las ranuras de implementación de este modo es beneficioso porque permite probar la nueva versión antes de que entre en producción e implementarla después en producción sin tiempo de inactividad.

Uso de registros de diagnósticos

Una forma eficiente de supervisar la aplicación es usar registros de diagnóstico para ver el registro de diagnóstico en directo de la aplicación web. Puedes incluso insertar los registros en la ventana de la consola. Para ello, ejecuta el siguiente comando en la Azure Cloud Shell:

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

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

Configuración de alertas y supervisión

[Azure Monitor Application Insights](#) proporciona otra poderosa forma de dar seguimiento a las aplicaciones. Esta herramienta de supervisión facilita información sobre tu aplicación, como cuántos visitantes la han usado, el número de excepciones que se han producido y el punto del código en el que ocurrieron. A diferencia de los registros de diagnóstico, Application Insights requiere una tarifa nominal.

Vamos a configurar Application Insights.

1. Ve al portal de Azure y a la aplicación web que hospeda la aplicación de .NET Core.
2. En la barra de menús, haz clic en **Application Insights**.
3. Selecciona **Crear recurso nuevo**.
4. Escribe un nombre y selecciona una ubicación para la instancia de Application Insights.

5. Haz clic en **Aceptar**. Application Insights se implementará y empezará a recopilar datos para la aplicación.

Ahora necesitas configurar la aplicación para que envíe datos a Application Insights.

6. En el menú Visual Studio Code, selecciona **Proyecto > Agregar telemetría de Application Insights...**
7. Esto abrirá el asistente de Application Insights. Inicia sesión con tu cuenta de Azure.
8. Selecciona un plan de precios de Application Insights.
9. Haz clic en **Registro**. Esto automáticamente agrega todo al proyecto de .NET Core y crea el recurso de Application Insights en Azure.
10. Desarrolla el proyecto e inserta los cambios en GitHub para que puedan implementarse en la aplicación web. Una vez completada la implementación, la aplicación enviará datos a Application Insights.

De manera predeterminada, Application Insights realiza detección inteligente. Esta característica detecta si algo va mal, como un aumento repentino de las solicitudes con error o si la aplicación funciona con una lentitud inusual, y te alerta de ello. 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 8-18.



NAME	CONDITION	LAST ACTIVE
MYSTHUBTEST (COMPONENTS)		
Failed requests per second	Failed Requests per Second > 1	Never

Figura 8-18

11. Para comprobar si Application Insights funciona correctamente, ve al portal de Azure, busca el recurso de Application Insights y selecciónalo. 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 la cantidad de solicitudes del servidor y de solicitudes fallidas. Deberías ver algunos datos, lo que significa que Application Insights está funcionando.

Escalado de la aplicación web

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

Web Apps tiene un elemento de menú denominado **Escalar horizontalmente**, como se muestra en la Figura 8-19. Puedes usarlo para ampliar la escala manual o automáticamente. Escalar horizontalmente significa agregar más instancias de la aplicación para gestionar la carga.

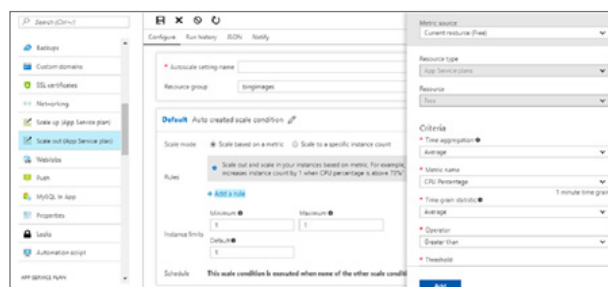


Figura 8-19

Agregar una capa de sockets seguros

Cuando una 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 medidas de seguridad más importantes que puedes tomar. El motivo es que, sin HTTPS, los intrusos podrían ver el tráfico entre tus recursos y utilizar esta información con fines malintencionados, como iniciar sesión en tu aplicación. Además, HTTPS es obligatorio para características avanzadas como los [service workers](#).

Suministrar tráfico a tu aplicación web a través de la capa de sockets seguros (SSL) es posible importando un certificado SSL en Web Apps y enlazándolo con uno de tus nombres de dominio personalizados. Puedes importar tu propio certificado SSL, o comprar uno nuevo a través de [Certificados de Azure App Service](#). Este servicio facilita la compra y validación de certificados. 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.

Notificar a los usuarios sobre nuevas versiones

Tu empresa puede aprovecharse de que los usuarios conozcan sus nuevos lanzamientos de productos. Al ampliar el proceso de integración/entrega continua (CI/CD) en las versiones de Azure, puedes usar un flujo de trabajo de Logic Apps para gestionar la comunicación en redes sociales, como la difusión de tweets o las publicaciones con notas de versión.

09 /

Resumen y recursos

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

Ya has descubierto que Azure tiene servicios para casi todos los casos, por lo que puede serte útil independientemente del lenguaje de programación que uses o de la plataforma para la que escribas las aplicaciones. Esperamos que sigas consultando este libro electrónico para familiarizarte un poco más con la amplia gama de servicios de Azure y para determinar cuáles se ajustan mejor a tus necesidades.

Gracias a la amplia variedad de soluciones preconfiguradas en Azure, se acabaron los días en que tenías que escribir complicadas secuencias de código. Libérate de trabajo para concentrarte en las cosas que importan aprovechando todo lo que Azure tiene que ofrecerte.

Sigue aprendiendo con Azure

Con tu [cuenta gratuita de Azure](#) podrás conseguir todo esto y no te cobraremos nada hasta que decidas mejorar tu cuenta:

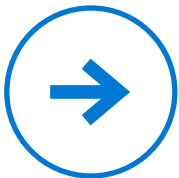
- 12 meses de servicios populares gratis.
- Un crédito de 200 USD para explorar cualquier servicio de Azure durante 30 días.
- Más de 25 servicios siempre gratuitos.
- [Empieza a usar Azure](#): mira estos breves tutoriales sobre cómo utilizar Azure y empezar a crear proyectos ya mismo. También puedes unirme a nuestro [seminario web semanal](#), que ofrece demostraciones de los aspectos básicos de Azure y facilita un acceso continuo a los expertos.
- [Microsoft Learn](#): aprende nuevas capacidades y descubre el potencial de los productos de Microsoft con orientación paso a paso. Inicia hoy tu viaje explorando nuestras vías y módulos de aprendizaje.
- [Azure Friday](#), que ofrece un vistazo a los servicios y características de Azure con el equipo de ingenieros de Microsoft.
- [Azure.Source](#): mantente al día sobre lo que ocurre en Azure, incluidas noticias y actualizaciones, el contenido de preview y lo que está generalmente disponible.
- [Sugerencias y trucos de Azure](#): explora una colección de ideas útiles para aumentar tu productividad con Azure.



Multitud de recursos gratuitos

Además de esta guía, hay muchos otros recursos gratuitos para Azure, incluidos:

- [Aprende Azure durante las comidas de un mes](#): un método práctico para aprender Azure desde cero a la hora de comer durante un mes.
- [Libro de instrucciones de la computación sin servidor de Azure](#): e-book sobre todos los aspectos sin servidor de Azure.
- [Diseño de sistemas distribuidos](#): e-book sobre la creación de aplicaciones en contenedores, con laboratorios prácticos de Azure Kubernetes Service.
- [Conteneriza tus aplicaciones con Docker y Kubernetes](#): guía práctica para trasladar aplicaciones al cloud con Docker y Kubernetes.
- [Guía de NoSQL con Azure Cosmos DB](#): e-book sobre cómo crear aplicaciones esenciales con gran capacidad de respuesta con Azure Cosmos DB.
- [DevOps eficaz](#): guía práctica sobre cómo mejorar la colaboración entre equipos, promover un uso eficiente de las herramientas y usar los conceptos de DevOps.
- [Guía para desarrolladores del IoT](#): e-book que facilita información general sobre los servicios del IoT de Azure y presenta una introducción en este ámbito.
- [Azure para arquitectos](#): guía completa para arquitectos en Azure.
- [Guía de creación de aplicaciones de IA para desarrolladores](#): guía práctica para crear tu primer bot inteligente con la IA.
- [La disrupción por diseño](#): inspiración y orientación sobre cómo son posibles los cambios transformadores y cómo conseguirlos.
- [Guía práctica de Microsoft Azure IaaS](#): consejos y prácticas recomendadas sobre cómo migrar sistemas on-premises al cloud con Azure.
- [Estrategia de cloud para la empresa](#): método demostrados de estrategias para trasladar tu empresa a la computación en el cloud.
- [Aspectos básicos de la migración al cloud](#): e-book sobre cómo simplificar tu ruta al cloud mientras minimizas los riesgos y las perturbaciones en tu negocio.
- [Sacar el máximo partido al cloud en cualquier lugar](#): e-book que se centra en el desarrollo unificado y las prácticas de modernización en entornos híbridos.



<http://www.azure.com/gratis>

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 Michael en Twitter [@mbcrump](#), sigue este blog en <https://www.michaelcrump.net/>, o sigue una publicación reciente de la serie [Consejos y trucos de Azure](#).



Barry Luijbregts es un arquitecto de software independiente, desarrollador apasionado del cloud y autor de cursos de Pluralsight.

Puedes comunicarte con Barry en Twitter [@AzureBarry](#) o mediante su sitio web en <https://www.azurebarry.com/>.

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

Copyright © 2019 de Microsoft Corporation. Todos los derechos reservados.
El contenido de este libro no puede reproducirse ni transmitirse de ninguna forma 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 necesitas ayuda en relación con este libro, envía un correo electrónico a Microsoft Press Support a msspinput@microsoft.com. Danos tu opinión sobre este libro en esta [encuesta](#).

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 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.

