# **Introducción**

Completado

100 XP

* 1 minuto

El término *informática sin servidor* se usa para describir un entorno de ejecución que se configura y administra de manera automática. El cliente tan solo debe escribir código o conectar y configurar los componentes en un editor visual y, después, especificar las acciones que desencadenan la funcionalidad, como un temporizador o una solicitud HTTP. Lo mejor de todo es que únicamente se paga en función del uso real del código y que no hay que preocuparse de las interrupciones, dado que el código puede hacer un escalado instantáneo para satisfacer la demanda.

Tailwind Traders es una empresa tradicional dedicada a la distribución de materiales de construcción. Vende mucho por Internet, pero cree que hay varios aspectos de su sitio web de comercio electrónico que se pueden mejorar. Por ejemplo, quiere que la información en tiempo real del inventario en línea sea más precisa para los clientes que desean visitar su almacén local para comprar un artículo. La empresa también quiere desarrollar un programa de fidelización de clientes que les permita responder de forma más proactiva a los usuarios que han tenido una experiencia negativa.

Tailwind Traders sospecha que la informática sin servidor puede ayudar a proporcionar estos servicios, pero necesita ayuda para saber qué soluciones de Azure son adecuadas para sus escenarios empresariales.

En este módulo, obtendrá información sobre dos soluciones informáticas sin servidor en Azure: Azure Functions y Azure Logic Apps. Verá lo que son, aprenderá en qué se diferencian y sabrá cuándo elegir una u otra.

# **Identificación de las opciones de producto**

Completado

100 XP

* 5 minutos

Vea este vídeo para obtener información general sobre la informática sin servidor.

Tal como se indica en el vídeo, la informática sin servidor es un entorno de ejecución hospedado en la nube que ejecuta código, pero abstrae el entorno de hospedaje subyacente. El término *informática sin servidor* es poco apropiado, ya que, al fin y al cabo, *hay* un servidor (o un grupo de servidores) que ejecuta el código o la funcionalidad.

La idea clave es que el cliente no es responsable de la configuración o el mantenimiento del servidor. No tiene que preocuparse de las interrupciones ni de escalarlo cuando hay un incremento en la demanda. El proveedor de la nube se encarga de todo el mantenimiento y el escalado.

El cliente crea una instancia del servicio y, después, agrega su código. No se requiere, ni siquiera se permite, ningún mantenimiento ni configuración de la infraestructura. Las aplicaciones sin servidor se configuran para responder a eventos. Estos podrían ser un punto de conexión REST, un temporizador periódico o incluso un mensaje recibido de otro servicio de Azure. La aplicación sin servidor se ejecuta solo cuando la desencadena un evento. La escalabilidad y el rendimiento se controlan automáticamente, y solo se facturan los recursos que se usan. Ni siquiera es necesario reservar recursos.

La informática sin servidor suele utilizarse para controlar los escenarios de *back-end*. En otras palabras, la informática sin servidor es responsable de enviar mensajes de un sistema a otro o de procesar mensajes enviados desde otros sistemas. No se usa para sistemas orientados al usuario, sino que funciona en segundo plano.

En este módulo, trataremos dos servicios de informática sin servidor de Azure: Azure Functions y Azure Logic Apps.

## **Azure Functions**

Con el servicio [Azure Functions](https://azure.microsoft.com/services/functions/), puede hospedar un único método o función mediante un lenguaje de programación popular en la nube que se ejecuta en respuesta a un evento. Un ejemplo de un evento podría ser una solicitud HTTP, un mensaje nuevo en una cola o un mensaje en un temporizador.

Por su naturaleza atómica, Azure Functions puede servir para muchos propósitos en el diseño de una aplicación. Las funciones se pueden escribir con muchos lenguajes de programación comunes, como C#, Python, JavaScript, Typescript, Java y PowerShell.

Azure Functions se escala automáticamente, y los cargos se acumulan solo cuando se desencadena una función. Estas características convierten a Azure Functions en una elección sólida cuando la demanda es variable. Por ejemplo, podría recibir mensajes de una solución de IoT que supervisa una flota de vehículos de reparto. Probablemente llegarán más datos durante el horario comercial. Azure Functions se puede escalar horizontalmente para adaptarse a esas horas de más trabajo.

Una función de Azure es un entorno sin estado. Una función se comporta como si se reiniciara cada vez que responde a un evento. Esta característica resulta muy conveniente para procesar los datos entrantes. Y si el estado es necesario, la función se puede conectar a una cuenta de almacenamiento de Azure.

Azure Functions puede realizar tareas de orquestación mediante una extensión llamada Durable Functions, que permite a los desarrolladores encadenar funciones al tiempo que se mantiene el estado.

La solución Azure Functions es ideal si le preocupa solo el código que ejecuta el servicio y no la infraestructura o la plataforma subyacente. Azure Functions se usa con más frecuencia cuando es necesario realizar un trabajo en respuesta a un evento. Esto suele realizarse mediante una solicitud REST, un temporizador o un mensaje de otro servicio de Azure, y cuando ese trabajo puede completarse rápidamente, en segundos o en menos tiempo.

## **Azure Logic Apps**

[Logic Apps](https://azure.microsoft.com/services/logic-apps/) es una plataforma de desarrollo de poco código o sin código hospedada como un servicio en la nube. El servicio le ayuda a automatizar y organizar tareas, procesos empresariales y flujos de trabajo cuando tiene que integrar aplicaciones, datos, sistemas y servicios en empresas u organizaciones. Logic Apps simplifica el diseño y la creación de soluciones escalables en la nube, en el entorno local o en ambos. Esta solución abarca la integración de aplicaciones, la integración de datos, la integración de sistemas, la integración de aplicaciones empresariales (EAI) y la integración de negocio a negocio (B2B).

Azure Logic Apps está diseñado en un entorno web y puede ejecutar una lógica que los servicios de Azure desencadenan sin escribir ningún código. Las aplicaciones se pueden compilar vinculando desencadenadores con acciones mediante conectores. Un desencadenador es un evento (como un temporizador) que hace que una aplicación se ejecute, que un mensaje nuevo se envíe a una cola o que se emita una solicitud HTTP. Una acción es una tarea o paso que se puede ejecutar. Hay acciones lógicas, como las que encontraría en la mayoría de los lenguajes de programación. Entre los ejemplos de acciones se incluyen trabajar con variables, instrucciones de decisión y bucles, y tareas que analizan y modifican datos.

Para crear soluciones de integración empresarial con Azure Logic Apps, se puede elegir entre una galería creciente de más de 200 conectores. La galería incluye servicios como Salesforce, SAP, Oracle DB y recursos compartidos de archivos.

Si no encuentra la acción o el conector que necesita, puede crear su propio conector mediante código personalizado.

## **¿Cuál es la diferencia entre estos servicios?**

Puede llamar a Azure Functions desde Azure Logic Apps y viceversa. La principal diferencia entre los dos servicios es su intención. Azure Functions es un servicio informático sin servidor, y Azure Logic Apps está diseñado para ser un servicio de orquestación sin servidor. Aunque puede usar Azure Functions para orquestar un proceso empresarial de larga duración que implique varias conexiones, su caso de uso principal no era ese cuando se diseñó.

Además, los dos servicios tienen un precio diferente. Los precios de Azure Functions se basan en el número de ejecuciones y en el tiempo ejecución de cada una. Los precios de Logic Apps se basan en el número de ejecuciones y el tipo de conectores que se usan.

# **Análisis de los criterios de decisión**

Completado

100 XP

* 5 minutos

Con dos opciones sin servidor posibles, puede ser difícil saber cuál es la más conveniente para el trabajo que debe realizarse. En esta unidad, analizaremos los criterios que emplean los expertos al elegir qué servicio sin servidor satisface mejor a una determinada necesidad empresarial. Comprender estos criterios también puede ayudar a entender mejor las diferencias entre los productos.

## **¿Necesita realizar una orquestación entre API conocidas?**

Como mencionamos anteriormente, Azure Logic Apps se diseñó pensando en la orquestación, desde el configurador visual basado en web hasta el modelo de precios. Logic Apps es excelente a la hora de conectar una gran variedad de servicios distintos mediante sus API para pasar y procesar los datos a través de los muchos pasos de un flujo de trabajo.

Es posible crear el mismo flujo de trabajo con Azure Functions, pero podría tardar una cantidad considerable de tiempo en averiguar a qué API debe llamar y cómo llamarlas. Azure Logic Apps ya sabe cómo tratar estas llamadas API, por lo que basta con suministrar algunos detalles para que las llamadas API necesarias se abstraigan.

## **¿Necesita ejecutar algoritmos personalizados o realizar análisis y búsquedas de datos especiales?**

Con Azure Functions, puede usar la expresividad completa de un lenguaje de programación en una forma compacta. Esto le permite crear de manera concisa algoritmos complejos u operaciones de búsqueda y análisis de datos. El cliente es el responsable de mantener el código, controlar las excepciones de manera resistente, etc.

Aunque Azure Logic Apps puede ejecutar la lógica (bucles, decisiones, etc.), si tiene una orquestación de lógica intensiva que requiera un algoritmo complejo, la implementación de ese algoritmo podría ser más detallada y visualmente abrumadora.

## **¿Tiene tareas automatizadas escritas en un lenguaje de programación imperativo?**

Si ya tiene la orquestación o la lógica de negocios expresada en C#, Java, Python u otro lenguaje de programación popular, podría ser más fácil trasladar el código al cuerpo de una aplicación de funciones de Azure Functions que volver a crearlo mediante Azure Logic Apps.

## **¿Prefiere un flujo de trabajo visual (declarativo) o escribir código (imperativo)?**

En última instancia, la elección se reduce a si prefiere trabajar en un entorno declarativo o en uno imperativo. Los desarrolladores que ya tienen experiencia con un lenguaje de programación imperativo pueden preferir adoptar una postura imperativa con respecto a la automatización y la orquestación. Los profesionales de TI y los analistas de negocios pueden preferir trabajar en un entorno (declarativo) sin código o de poco código y más visual.

# **Uso de Azure Functions**

Completado

100 XP

* 3 minutos

Los datos de cada producto que se vende en Tailwind Traders se empaquetan como un mensaje JSON y se envían a un centro de eventos. El centro de eventos distribuye el mensaje JSON a los suscriptores, lo que permite notificar a los distintos sistemas.

Tailwind Traders quiere actualizar su sitio de comercio electrónico para incluir el seguimiento del inventario en tiempo real. Actualmente, el sitio web actualiza la disponibilidad del producto todas las noches a las 2:00 a. m. Un servicio de Windows escrito en C# contiene toda la lógica necesaria para:

* Recuperar los mensajes
* Analizar JSON
* Realizar una búsqueda en varias bases de datos para buscar información adicional del producto
* Puede enviar notificaciones al departamento de compras para que pueda reponer las cantidades que se encuentran por debajo de determinados niveles.

El servicio de Windows se ejecuta en una máquina virtual hospedada en Azure.

La mayoría del tiempo, este sistema funciona correctamente. No obstante, hay una demanda elevada de algunos productos, mientras que de otros productos hay pocas unidades en los almacenes. Varias veces al día, hay clientes que van a una tienda para recoger un artículo del que ya no quedan existencias.

En lugar de ejecutar el algoritmo cada noche, la empresa quiere que la actualización del inventario se realice cada vez que se adquiere un producto.

## **¿Qué servicio debe elegir?**

Como el equipo de desarrolladores de Tailwind Traders ya tiene la lógica escrita en C#, tendría sentido copiar el código C# pertinente del servicio de Windows y trasladarlo a una función de Azure. Los desarrolladores tendrían que enlazar la función para que se desencadene cada vez que aparezca un mensaje nuevo en una determinada cola.

## **¿Por qué no elegir Azure Logic Apps?**

Es posible implementar la misma lógica en Azure Logic Apps. No obstante, dado que el equipo ya ha invertido tiempo en crear el servicio en C#, puede usar el mismo código en una función de Azure.

# **Uso de Azure Logic Apps**

Completado

100 XP

* 2 minutos

Tras una compra, Tailwind Traders envía a sus clientes una invitación aleatoriamente para participar en una encuesta de satisfacción del cliente. Actualmente, los resultados de dicha encuesta se agregan, se calcula su promedio y se plasman en un gráfico. Pero el departamento de atención al cliente, con una actitud proactiva, quiere ponerse en contacto con los clientes que proporcionan puntuaciones bajas y dejan comentarios con una opinión negativa.

Lo ideal sería que las puntuaciones de satisfacción del cliente negativas desencadenasen un flujo de trabajo de retención de clientes. En primer lugar, se puede generar un análisis de opinión en función de los comentarios libres, se enviaría un correo electrónico al cliente con una disculpa y un código de descuento, y este mensaje se enrutaría al servicio de atención al cliente de Dynamics 365 para que pudiera programar un correo electrónico de seguimiento.

Desafortunadamente, no hay ningún desarrollador de Tailwind Traders disponible para realizar este proyecto. Pero el equipo de atención al cliente trabaja con varios profesionales de la nube y de TI que pueden dar una solución.

## **¿Qué servicio debe elegir?**

En este escenario, es probable que Azure Logic Apps sea la mejor solución. Un profesional de la nube o de TI puede usar los conectores existentes para realizar un análisis de opinión mediante el conector de Azure Cognitive Services, enviar un correo electrónico con el conector de Office 365 Outlook y crear un registro y un correo electrónico de seguimiento con el conector de servicio al cliente de Dynamics 365.

Dado que Azure Logic Apps es un servicio sin código o de poco código, no se necesitan desarrolladores. Un profesional de la nube o de TI debe pueden compilar y dar soporte a este flujo de trabajo.

## **¿Por qué no elegir Azure Functions?**

Aunque es posible compilar toda la solución mediante Azure Functions, este enfoque supondría todo un desafío si no se puede asignar el proyecto a ningún desarrollador de software.

Es un escenario ideal para Azure Logic Apps. Ya existen conectores para cada uno de los pasos descritos en el flujo de trabajo. Un desarrollador tendría que llevar a cabo una gran labor de investigación, desarrollo y pruebas para crear una solución que use todos estos sistemas de software diferentes.