**Escolher a melhor tecnologia sem servidor do Azure para seu cenário empresarial**

## Pré-requisitos

* Noções básicas dos conceitos de orquestração, fluxos de trabalho e API (interfaces de programação de aplicativo)
* Familiaridade de alto nível com produtos relevantes da Microsoft, como Dynamics 365 e Office 365

# Introdução

A computação sem servidor é um termo usado para descrever um ambiente de execução configurado e gerenciado para você. Você só precisa especificar o que deseja que aconteça escrevendo código ou conectando e configurando componentes em um editor visual e, então, especificar as ações que disparam sua funcionalidade, como um temporizador ou uma solicitação HTTP. O melhor é que você não precisará se preocupar com uma interrupção; seu código pode ser dimensionado instantaneamente para atender à demanda, e você só paga pelo uso real dele.

A Tailwind Traders é uma varejista tradicional com lojas físicas que obteve sucesso em vendas online. A empresa vê várias oportunidades para melhorar seu site de comércio eletrônico. Por exemplo, ela quer fornecer informações de inventário mais precisas em tempo real e online a clientes que desejam visitar a loja local para comprar um item. A empresa também quer responder de forma mais proativa aos clientes que tiveram uma experiência negativa, oferecendo um novo programa de retenção de clientes.

A Tailwind Traders acredita que a computação sem servidor pode ajudar a fornecer esses serviços, mas precisa de ajuda para entender quais soluções do Azure são ideais para seus cenários empresariais.

Neste módulo, você aprenderá sobre duas soluções de computação sem servidor no Azure: Azure Functions e Aplicativos Lógicos do Azure. Você aprenderá o que são essas soluções, como elas se diferenciam e quando escolher uma ou a outra.

## Objetivos de aprendizagem

Depois de concluir este módulo, você poderá:

* Escolher a tecnologia de computação sem servidor que melhor atende ao seu cenário empresarial.

## Pré-requisitos

* Compreensão do conceito de orquestração e fluxos de trabalho
* Compreensão do conceito de API (interface de programação de aplicativo)
* Familiaridade de alto nível com produtos relevantes da Microsoft, como Dynamics 365 e Office 365

# Identificar as opções de produto

# 

# Como mencionado no vídeo, a computação sem servidor é um ambiente de execução hospedado na nuvem que executa o código, mas abstrai o ambiente de hospedagem subjacente. O termo *computação sem servidor* é um nome inadequado. Afinal, *há* um servidor (ou um grupo de servidores) que executa seu código ou a funcionalidade desejada.

**A ideia principal é que você não seja responsável por configurar nem por manter esse servidor. Você não precisará se preocupar em dimensioná-lo quando houver aumento da demanda, nem com interrupções. O fornecedor de nuvem cuida de todas as questões de manutenção e dimensionamento para você.**

**Você cria uma instância do serviço e adiciona seu código. Nenhuma configuração ou manutenção da infraestrutura é necessária, nem mesmo permitida. Você configura os aplicativos sem servidor para responder a eventos. Um evento pode ser um ponto de extremidade REST, um temporizador periódico ou até mesmo uma mensagem recebida de outro serviço do Azure. Nesse caso, o aplicativo sem servidor é executado somente quando disparado por um evento. O dimensionamento e o desempenho são administrados automaticamente, e você é cobrado apenas pelos recursos que utiliza. Você nem precisa reservar recursos.**

**A computação sem servidor é normalmente usada para lidar com cenários de *back-end*. Em outras palavras, a computação sem servidor é responsável por enviar mensagens de um sistema para outro ou processar mensagens que foram enviadas de outros sistemas. Essa computação não é usada para sistemas voltados ao usuário; em vez disso, ela funciona em segundo plano.**

**Neste módulo, abordaremos dois serviços de computação sem servidor do Azure: Azure Functions e Aplicativos Lógicos do Azure.**

## Funções do Azure

Com o serviço do [Azure Functions](https://azure.microsoft.com/services/functions/), você pode hospedar um único método ou função usando uma linguagem de programação popular na nuvem que é executada em resposta a um evento. Um exemplo de evento pode ser uma solicitação HTTP, uma nova mensagem em uma fila ou uma mensagem em um temporizador.

Devido à sua natureza atômica, o Azure Functions pode atender a muitas finalidades no design de um aplicativo. As funções podem ser escritas em muitas linguagens de programação comuns, como C#, Python, JavaScript, Typescript, Java e PowerShell.

O Azure Functions é dimensionado automaticamente, e as cobranças são geradas apenas quando uma função é disparada. Essas qualidades tornam o Azure Functions uma escolha sólida quando a demanda é variável. Por exemplo, vamos supor que você receba mensagens de uma solução de IoT (Internet das Coisas) que monitora uma frota de veículos de entrega. Provavelmente, a maioria dos dados é recebida durante o horário comercial. É possível escalar horizontalmente o Azure Functions para acomodar esses horários de pico.

Uma função do Azure é um ambiente sem estado. Essa função se comporta como se fosse reiniciada sempre que responde a um evento. Esse recurso é ideal para o processamento de dados de entrada. E, se o estado for necessário, a função poderá ser conectada a uma conta de armazenamento do Azure.

O Azure Functions pode executar tarefas de orquestração usando uma extensão chamada Durable Functions, que permite aos desenvolvedores encadear funções mantendo o estado.

A solução Azure Functions é ideal quando você está preocupado apenas com o código que está executando seu serviço, e não com a plataforma ou infraestrutura subjacente. O Azure Functions geralmente é usado quando você precisa realizar um trabalho em resposta a um evento. Isso geralmente é feito por meio de uma solicitação REST, um temporizador ou uma mensagem de outro serviço do Azure e quando esse trabalho pode ser concluído rapidamente, em poucos segundos ou menos.

## **Aplicativos Lógicos do Azure**

Os [Aplicativos Lógicos](https://azure.microsoft.com/services/logic-apps/) são uma plataforma de desenvolvimento de código baixo/sem código hospedada como um serviço de nuvem. O serviço ajuda a automatizar e orquestrar tarefas, processos empresariais e fluxos de trabalho quando você precisa integrar aplicativos, dados, sistemas e serviços em empresas ou organizações. Os Aplicativos Lógicos simplificam a maneira como você projeta e constrói soluções escalonáveis, seja na nuvem, no local ou em ambos. Essa solução abrange as integrações de aplicativos, dados, sistemas, EAI (integração de aplicativos empresariais) e B2B (entre empresas).

Os Aplicativos Lógicos do Azure são projetados em um designer baseado na Web e podem executar a lógica disparada pelos serviços do Azure sem haver necessidade de codificação. Você cria um aplicativo vinculando gatilhos a ações com conectores. Um gatilho é um evento (por exemplo, um temporizador) que faz com que um aplicativo seja executado, uma nova mensagem seja enviada para uma fila ou uma solicitação HTTP seja gerada. Uma ação é uma tarefa ou etapa que pode ser executada. Existem ações lógicas como aquelas que você encontra na maioria das linguagens de programação. Exemplos de ações incluem trabalho com variáveis, loops e instruções de decisão e tarefas de análise e modificação de dados.

Para criar soluções de integração corporativa com Aplicativos Lógicos do Azure, você pode escolher em uma galeria crescente de mais de 200 conectores. A galeria inclui serviços como Salesforce, SAP, Oracle DB e compartilhamentos de arquivos.

Se você não encontrar a ação ou o conector de que precisa, poderá criar o seu usando código personalizado.

## **Quais são as diferenças entre esses serviços?**

Você pode chamar o Azure Functions de Aplicativos Lógicos do Azure e vice-versa. A principal diferença entre os dois serviços é a que eles se destinam. O Azure Functions é um serviço de computação sem servidor, enquanto os Aplicativos Lógicos do Azure se destinam a ser um serviço de orquestração sem servidor. Embora você possa usar o Azure Functions para orquestrar um processo empresarial de longa execução que envolva diversas conexões, esse não era o caso de uso principal dele quando foi projetado.

Os dois serviços também têm preços diferentes. O preço do Azure Functions é baseado no número de execuções e no tempo de execução de cada uma. O preço dos Aplicativos Lógicos é baseado no número de execuções e no tipo de conectores que elas utilizam.

# Analisar os critérios de decisão

# Com duas opções viáveis sem servidor, pode ser difícil saber qual é a melhor para o trabalho. Nesta unidade, analisaremos os critérios que os especialistas empregam ao escolher um serviço sem servidor para usar em determinada necessidade empresarial. O reconhecimento dos critérios também pode ajudar você a entender melhor as nuances entre os produtos.

## **Você precisa executar uma orquestração em APIs bem conhecidas?**

Como observamos antes, o serviço Aplicativos Lógicos do Azure foi projetado pensando em orquestração, do configurador visual baseado na Web ao modelo de preços. Os Aplicativos Lógicos se destacam na conexão de uma ampla gama de serviços diferentes por meio das APIs para passar e processar dados em várias etapas em um fluxo de trabalho.

É possível criar o mesmo fluxo de trabalho usando o Azure Functions, mas isso pode levar um tempo considerável para pesquisar quais APIs chamar e como chamá-las. Os Aplicativos Lógicos do Azure já modelaram essas chamadas à API para que você forneça apenas alguns dados, abstraindo os detalhes da chamada às APIs necessárias.

## **Você precisa executar algoritmos personalizados ou executar análises de dados especializadas e pesquisas de dados?**

Com o Azure Functions, você pode usar a expressividade total de uma linguagem de programação de forma compacta. Isso permite que criar algoritmos complexos de maneira concisa ou operações de pesquisa e análise de dados. Você seria responsável pela manutenção do código, manipulação de exceções de modo resiliente e assim por diante.

Embora os Aplicativos Lógicos do Azure possam executar lógica (loops, decisões e assim por diante), se você tiver uma orquestração de lógica intensiva que requeira um algoritmo complexo, implementar esse algoritmo poderá exigir maior detalhamento e parecer visualmente opressivo.

## **Você tem tarefas automatizadas escritas em uma linguagem de programação imperativa?** Se você já tem sua orquestração ou lógica de negócios expressa em C#, Java, Python ou em outra linguagem de programação popular, talvez seja mais fácil portar seu código para o corpo de um aplicativo de funções do Azure Functions do que recriá-lo usando os Aplicativos Lógicos do Azure.

## **Você prefere um fluxo de trabalho visual (declarativo) ou escrever código (imperativo)?** Em última análise, sua escolha se resume a definir se você prefere trabalhar em um ambiente declarativo ou imperativo. Os desenvolvedores que têm experiência com linguagem de programação imperativa podem preferir pensar na automação e orquestração com uma mentalidade imperativa. Os profissionais de TI e analistas de negócios podem preferir trabalhar em um ambiente mais visual de código baixo/sem código (declarativo).

# Usar o Azure Functions

Os dados sobre cada produto vendido na Tailwind Traders são empacotados como uma mensagem JSON e enviados a um hub de eventos. Esse hub distribui a mensagem JSON aos assinantes, o que permite que vários sistemas sejam notificados.

A Tailwind Traders deseja atualizar seu site de comércio eletrônico para incluir rastreamento de estoque em tempo real. Atualmente, o site atualiza a disponibilidade do produto à noite às 2h. Um serviço Windows escrito em C# contém toda a lógica necessária para:

* Recuperar as mensagens.
* Analisar o JSON.
* Executar uma pesquisa em vários bancos de dados para localizar informações adicionais do produto.
* Potencialmente, enviar notificações ao departamento de compras para que eles possam solicitar novamente as quantidades que fiquem abaixo de certos níveis.

O serviço Windows é executado em uma máquina virtual hospedada no Azure.

Na maioria das vezes, esse sistema funciona bem. No entanto, alguns produtos estão em alta demanda e outros produtos são mantidos em pequenas quantidades em cada loja. Várias vezes por dia, os clientes vão até a loja para comprar um item e descobrem que ele não está mais em estoque.

Em vez de executar o algoritmo todas as noites, a empresa deseja executar o atualizador de estoque cada vez que um produto é comprado.

## **Por que não escolher Aplicativos Lógicos do Azure?**

É possível implementar a mesma lógica nos Aplicativos Lógicos do Azure. No entanto, como a equipe já investiu tempo na construção do serviço em C#, ela pode usar o mesmo código em uma função do Azure.

# Usar os Aplicativos Lógicos do Azure

A Tailwind Traders envia a seus clientes um convite para participar de uma pesquisa de satisfação aleatoriamente após uma compra. Atualmente, os resultados da satisfação do cliente são agregados, calculados e colocados em gráfico. No entanto, seu departamento de atendimento ao cliente vê uma oportunidade de alcançar de forma proativa os clientes que fornecem notas baixas e deixam comentários com um sentimento negativo.

Idealmente, as pontuações negativas de satisfação dos clientes disparariam um fluxo de trabalho de retenção. Primeiro, uma análise de sentimentos seria gerada com base nos comentários em forma livre, um email seria enviado ao cliente com um pedido de desculpas e um código de cupom, e a mensagem seria encaminhada à equipe de atendimento ao cliente do Dynamics 365, para que ela pudesse agendar um email de acompanhamento.

Infelizmente, nenhum recurso de desenvolvedor da Tailwind Traders está disponível para assumir esse projeto. Mas a equipe de atendimento ao cliente trabalha com vários profissionais de nuvem e TI que talvez consigam criar uma solução.

## **Qual serviço escolher?**

Nesse cenário, os Aplicativos Lógicos do Azure são provavelmente a melhor solução. Um profissional de TI ou de nuvem poderia empregar conectores existentes para executar uma análise de sentimentos usando o conector de Serviços Cognitivos do Azure, enviar um email usando o conector do Outlook para Office 365 e criar um registro e email de acompanhamento usando o conector de atendimento ao cliente do Dynamics 365.

Como os Aplicativos Lógicos do Azure são um serviço de código baixo/sem código, nenhum desenvolvedor é necessário. Um profissional de TI ou de nuvem deve conseguir criar e dar suporte a esse fluxo de trabalho.

## **Por que não escolher o Azure Functions?**

Embora seja possível criar a solução inteira usando o Azure Functions, essa abordagem poderá ser desafiadora se nenhum desenvolvedor de software puder se comprometer com esse projeto.

Esse é um cenário ideal para os Aplicativos Lógicos do Azure. Os conectores já existem para cada uma das etapas descritas no fluxo de trabalho. Seria necessário um pouco de pesquisa, desenvolvimento e teste para um desenvolvedor criar uma solução que utilizasse todos esses sistemas de software diferentes.

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

Texto, Aplicativo

Descrição gerada automaticamente

# Resumo

Neste módulo, queríamos ajudar a Tailwind Traders a escolher a tecnologia de computação sem servidor certa para os cenários empresariais da empresa.

Quando a empresa precisou criar uma solução que efetuasse pull da lógica de código de um serviço Windows em C# existente, nós a ajudamos a escolher o Azure Functions.

Quando ela precisou orquestrar um fluxo de trabalho para melhorar a retenção do cliente após uma experiência de compra negativa, nós a ajudamos a escolher os Aplicativos Lógicos do Azure.

Em ambos os casos, observamos como seria possível escolher o outro serviço de computação sem servidor. No entanto, procuramos ajudar a empresa a considerar os critérios de decisão que traçamos e escolher o serviço certo para o cenário.

Sem a computação sem servidor, a Tailwind Traders seria forçada a configurar e gerenciar a própria infraestrutura de computação para esses cenários empresariais. A equipe precisaria monitorar de perto os serviços para determinar se era necessário dimensioná-los. E provavelmente teria desperdiçado dinheiro no processo, com muitos ou poucos recursos de computação dedicados à solução.

Além disso, talvez ela precisasse projetar, escrever, testar e manter um código personalizado para obter resultados semelhantes.

Ao auxiliarmos a Tailwind Traders na seleção das soluções de computação sem servidor ideais, conseguimos implantar uma nova funcionalidade para ajudar a empresa a melhorar a satisfação do cliente com sua plataforma de comércio eletrônico.