

Guia de prova de conceito do Azure para desenvolvedores



Guia de prova de conceito do Azure para desenvolvedores

03 /

Introdução

- 4 O que é uma prova de conceito?
- 5 Sobre este guia
- 5 O que é o Microsoft Azure?

06 /

Capítulo 1: Guia de prova de conceito

- 6 Etapa 1: Definir a meta e os critérios de sucesso
- 6 Etapa 2: Definir o cronograma e o orçamento
- 7 Etapa 3: Definir o escopo do projeto de prova de conceito
- 8 Etapa 4: Criar uma arquitetura de alto nível
- 9 Etapa 5: Montar sua equipe
- 9 Etapa 6: Implementação e testes
- 9 Etapa 7: Quando sua prova de conceito estiver concluída

11 /

Capítulo 2: Projeto de exemplo – implementar uma aplicação Web usando Aplicativos Web Estáticos do Azure

- 13 Aplicativos Web Estáticos do Azure vs. servidor Web tradicional
- 13 Principais recursos de Aplicativos Web Estáticos do Azure
- 13 Caso de uso de Aplicativos Web Estáticos do Azure
- 14 Colocar em prática
- 21 Para saber mais

22 /

Capítulo 3: Projeto de exemplo – criar um chatbot inteligente

- 22 Principais recursos dos chatbots
- 24 Caso de uso dos chatbots
- 25 Colocar em prática

36 /

Capítulo 4: Uma visão geral do Azure para desenvolvedores

- 36 Introdução ao Microsoft Azure
- 36 Os benefícios do Azure
- 37 Assinatura do Azure
- 37 O que a conta gratuita do Azure inclui?

50 /

Capítulo 5: Mais aprendizagem e recursos

- 50 Aprender os conceitos básicos do Azure
- 50 Ferramentas necessárias para desenvolver seu projeto de prova de conceito para o Azure
- 51 Outros recursos úteis

52 /

Conclusão

© 2020 Microsoft Corporation. Todos os direitos reservados.

Este documento é fornecido “no estado em que se encontra”. As informações e as opiniões expressas neste documento, incluindo URLs e outras referências a sites da Internet, podem ser alteradas sem aviso prévio. Você assume o risco de utilização. Este documento não oferece a você direitos legais sobre a propriedade intelectual de produtos da Microsoft. Você poderá copiar e usar este documento para finalidades internas e de referência.

Introdução

Com o advento de novas tecnologias, muitas organizações estão embarcando em projetos de prova de conceito para aprender e explorar novos recursos, além de realizar avaliações de viabilidade de conceitos propostos. Uma prova de conceito é uma primeira etapa importante no cultivo de inovações nos negócios.

Em uma pesquisa recente (consulte Figura 1: Como ferramentas/plataformas são avaliadas e adotadas por desenvolvedores) realizada pela Packt Publishing, a criação de um projeto de prova de conceito é a escolha mais popular para os desenvolvedores quando se trata de avaliar e adotar novas ferramentas e plataformas.



Figura 1: Como ferramentas/plataformas são avaliadas e adotadas por desenvolvedores

O que é uma prova de conceito?

Uma prova de conceito é um exercício temporário com escopo, metas específicas mensuráveis e métricas de sucesso. O ideal é que se baseie na realidade dos negócios para que os resultados sejam significativos.

Benefícios dos projetos de prova de conceito

Um projeto de prova de conceito pode ser uma ferramenta valiosa para avaliar se uma tecnologia ou conceito potencial pode ser usado para atender aos requisitos de uma solução de negócios. Ele pode ajudar a identificar possíveis problemas técnicos e logísticos antes que o serviço seja implementado em um projeto base. Além disso, ele fornece insights oportunos sobre a tecnologia e, ao mesmo tempo, reduz os riscos, permitindo que as principais decisões sejam tomadas nos estágios iniciais do processo de desenvolvimento.

Os projetos de prova de conceito oferecem vários benefícios:

Experimentar novas tecnologias

À medida que as tecnologias evoluem, os projetos de prova de conceito permitem que as organizações descubram, aprendam e experimentem tecnologias inovadoras que podem ser usadas em futuros projetos. Sob as diretrizes de uma execução com escopo e prazo definidos, a equipe de desenvolvimento pode rapidamente intensificar o uso de novas tecnologias sem colocar em risco os projetos base em grande escala da empresa. O melhor de tudo é que o sucesso do projeto de prova de conceito pode ser adaptado em um projeto base no futuro. Como resultado, os projetos de prova de conceito incentivam a inovação.

Minimizar os riscos

Antes de embarcar em um projeto de alto risco, em grande escala e potencialmente caro, é recomendável minimizar os riscos e os custos realizando uma rápida validação das partes do projeto que são consideradas arriscadas. Por exemplo, uma equipe de projeto pode identificar e isolar uma tecnologia específica usada em um projeto em grande escala e sinalizá-la como arriscada devido à inexperiência da equipe de desenvolvimento com essa tecnologia. A equipe de projeto pode realizar uma rápida experimentação de viabilidade e avaliação dessa tecnologia ao analisar um pequeno subprojeto com orçamento e cronograma fixos. Os riscos são mitigados, pois os projetos de prova de conceito normalmente são executados em um ambiente controlado e de área restrita.

Sobre este guia

O objetivo do Guia de prova de conceito do Azure para desenvolvedores é fornecer ao leitor orientações sobre como planejar um projeto de prova de conceito bem-sucedido. Ele também fornecerá orientações sobre como começar a desenvolver na plataforma de nuvem do Microsoft Azure. Este guia foi criado para desenvolvedores e arquitetos que estão embarcando em sua primeira jornada rumo ao Azure.

A metodologia descrita no Capítulo 1: Guia de prova de conceito cria a base central necessária para planejar e executar um projeto de prova de conceito bem-sucedido.

No Capítulo 2: Projeto de exemplo – implementar uma aplicação Web usando Aplicativos Web Estáticos do Azure e no Capítulo 3: Projeto de exemplo – criar um chatbot inteligente, vamos mostrar dois projetos práticos que podem inspirá-lo a criar seus próprios projetos de prova de conceito.

No Capítulo 4: Uma visão geral do Azure para desenvolvedores, apresentamos a plataforma do Azure. Se você for novo no Azure, aprenderá a obter uma conta gratuita do Azure para iniciar sua jornada de prova de conceito no Azure. Você também aprenderá a escolher o modelo de nuvem e os serviços apropriados para começar a desenvolver seu projeto para o Azure.

Além disso, forneceremos alguns materiais de aprendizagem úteis no Capítulo 5: Mais aprendizagem e recursos.

O que é o Microsoft Azure?

O foco deste guia será o Microsoft Azure. O Azure é uma plataforma de nuvem que permite hospedar suas aplicações existentes com alterações mínimas, bem como desenvolver novas aplicações nativas de nuvem. Há um grande número de serviços do Azure prontos para uso que podem ser integrados às suas aplicações para aproveitar instantaneamente os novos recursos de nuvem, minimizando a necessidade de você mesmo desenvolver esses serviços.

Ao hospedar suas aplicações no Azure, você pode criar um Produto Mínimo Viável e, em seguida, escalar sua aplicação conforme a demanda dos clientes cresce. O Azure oferece a confiabilidade necessária para aplicações de alta disponibilidade, incluindo failover entre diferentes regiões. O portal do Azure permite gerenciar facilmente todos os seus serviços do Azure. Você também pode gerenciar seus serviços programaticamente usando APIs e modelos específicos aos serviços.

Vamos aprender mais sobre o Azure no Capítulo 4: Uma visão geral do Azure para desenvolvedores.

Com tanta coisa para abordar, vamos começar com o guia de prova de conceito.



Capítulo 1: Guia de prova de conceito

Etapa 1: Definir a meta e os critérios de sucesso

A maioria dos projetos de prova de conceito é controlada por resultados. Defina a meta que deseja atingir com seu projeto de prova de conceito e determine o que seria considerado uma execução bem-sucedida do projeto, declarando os critérios de sucesso.

Queremos provar se um conceito funciona ou não sem investir uma enorme quantidade de tempo e recursos. Se o projeto de prova de conceito for bem-sucedido e atingir os resultados esperados, a equipe do projeto poderá autorizar a execução das próximas etapas. No entanto, se a experimentação falhar, ela falhará dentro de orçamento, cronograma, escopo e conjunto de recursos fixos, o que representará custos mínimos considerando tudo. Portanto, o benefício de um projeto de prova de conceito é que ele permite à equipe do projeto determinar rapidamente se deseja insistir no conceito, abandoná-lo ou fazer escolhas alternativas. Ao analisar seu projeto (consulte Etapa 3: Definir o escopo do projeto de prova de conceito), lembre-se de que o objetivo da sua prova de conceito é ter êxito rapidamente (e, em contrapartida, falhar rapidamente) para que as decisões subsequentes possam ser tomadas em tempo hábil.

Etapa 2: Definir o cronograma e o orçamento

Após a definição da meta e dos critérios de sucesso para o projeto de prova de conceito, você definirá o cronograma e o orçamento.

Em nossa experiência, os projetos de prova de conceito têm o melhor resultado quando são condensados entre duas e quatro semanas. Isso fornece tempo suficiente para que o trabalho seja concluído sem a sobrecarga de casos de uso e matrizes de teste complexas.

Veja algumas dicas:

- Faça estimativas realistas do tempo necessário para concluir as tarefas na sua prova de conceito.
- Se você acredita que a prova de conceito levará mais de quatro semanas, procure reduzir o escopo para se concentrar nas metas de maior prioridade.
- Obtenha o apoio de todos os principais recursos e patrocinadores para o cronograma e o orçamento antes de continuar.

Agora que você determinou o cronograma e o orçamento, vamos passar para o escopo da prova de conceito.

Etapa 3: Definir o escopo do projeto de prova de conceito

É importante definir o escopo do seu projeto de prova de conceito antes de iniciar a implementação para evitar a lentidão do escopo. Saber antecipadamente que os recursos não ficarão disponíveis por tempo indeterminado define expectativas apropriadas para as partes interessadas da prova de conceito.

Dica: evite transformar um projeto de prova de conceito em um projeto de produção definindo seu escopo antes de iniciar o projeto. Depois, ao longo do projeto, garanta a adesão de todos os envolvidos ao escopo.

As seguintes perguntas-chave ajudarão você a determinar o escopo do seu projeto de prova de conceito:

- O que você deseja aprender ou alcançar com a sua prova de conceito?
- Quais são os critérios de sucesso?
- Quais workloads ou cenários serão abordados?

Dica: para garantir que seu projeto de prova de conceito possa ter o escopo definido e ser concluído em tempo hábil, mantenha os workloads e cenários com o menor tamanho possível.

- Que recursos devem estar disponíveis?
- Quem são os usuários e as equipes que validarão o resultado?
- Qual é a duração do projeto?

Dica: escolha um cronograma que se alinhe bem com o escopo planejado, como duas ou quatro semanas.

- Depois que a prova de conceito for concluída, o que acontecerá com os recursos que foram alocados durante o projeto? Você planeja descartar esses recursos?

Com o escopo definido, podemos criar uma arquitetura de alto nível para a prova de conceito.

Etapa 4: Criar uma arquitetura de alto nível

Tenha em mente que você está desenvolvendo uma prova de conceito para validar se um conceito é viável. Portanto, a fim de manter a meta do projeto alcançável dentro de um cronograma bem definido, ao criar sua arquitetura de alto nível, você deve decidir quais componentes essenciais farão parte da prova de conceito e quais componentes não essenciais devem ser excluídos dela.

Como regra geral, quanto mais componentes você incluir na sua prova de conceito, mais complicado o projeto se tornará e mais tempo levará para alcançar seus critérios de sucesso.

Como parte da sua arquitetura de alto nível, você também deverá escolher o modelo de nuvem apropriado para hospedar sua aplicação de prova de conceito. Você aprenderá “Como escolher o modelo de nuvem apropriado para seu projeto de prova de conceito do Azure” no Capítulo 4: Uma visão geral do Azure para desenvolvedores.

Como prática recomendada, seus projetos de prova de conceito devem sempre residir em uma assinatura do Azure separada da produção. Idealmente, uma assinatura de Desenvolvimento/Teste deve ser usada para manter o custo baixo. Você aprenderá mais sobre a assinatura do Azure na seção Getting started with Microsoft Azure no Capítulo 4: Uma visão geral do Azure para desenvolvedores.

Dica: a perfeição não é necessária. É tentador tentar arquitetar uma prova de conceito perfeita, que espelhará o produto eventual se ele for distribuído para produção. No entanto, para um projeto de prova de conceito, isso seria contraproducente. Quanto mais perto você tentar chegar da perfeição, mais tempo e esforço terá que empregar do início ao fim. Isso seria indesejável. Lembre-se de que o objetivo da prova de conceito é provar rapidamente um determinado conceito bem definido. É permitir que você tome as decisões corretas em tempo hábil. Portanto, o foco deve ser sempre a seleção das menores dependências essenciais e workloads associados que atendam a metas mensuráveis específicas, para ajudar a garantir uma vitória rápida.

Com o plano arquitetônico em vigor, você está pronto para montar sua equipe para o projeto de prova de conceito.

Etapa 5: Montar sua equipe

Para uma pequena prova de conceito pontual, simples, não crítica e experimental, uma equipe de uma pessoa pode ser suficiente. No entanto, para a maioria dos projetos típicos de prova de conceito em que os resultados são críticos e podem influenciar a decisão de um projeto maior, você deve identificar os membros obrigatórios da equipe e o compromisso necessários para oferecer suporte à sua prova de conceito. A equipe que você está montando deve refletir o escopo do seu projeto.

Etapa 6: Implementação e testes

Com a meta, o cronograma, o orçamento e o escopo definidos e a equipe montada, você pode começar a implementar seu projeto de prova de conceito com base na arquitetura de alto nível. Para maximizar o sucesso da execução, siga processos modernos de DevOps com desenvolvimento e testes iterativos em toda a sua implementação.

Etapa 7: Quando sua prova de conceito estiver concluída

Quando sua prova de conceito estiver concluída, avalie se você cumpriu os critérios de sucesso que definiu na Etapa 1: Definir a meta e os critérios de sucesso.

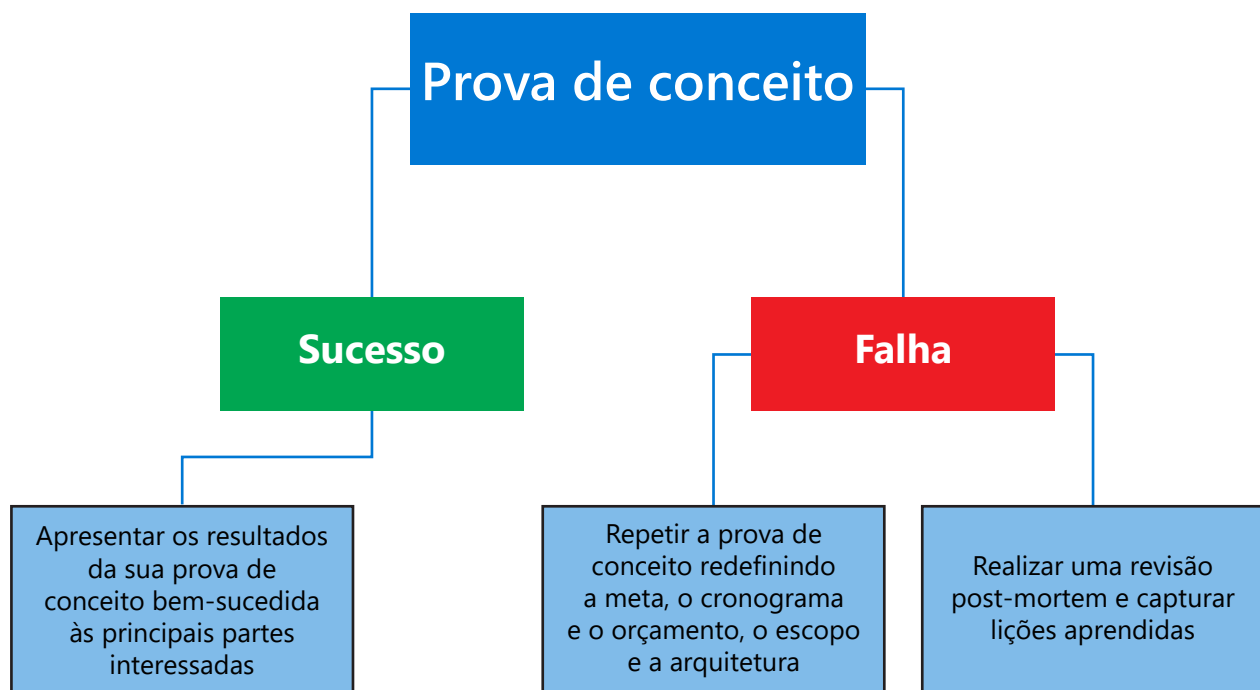


Figura 2 - Avaliar sua prova de conceito

Se a sua prova de conceito for bem-sucedida

- Apresente seus resultados bem-sucedidos às principais partes interessadas.
Dica: ao apresentar sua prova de conceito bem-sucedida aos principais tomadores de decisão, tente extrapolar o sucesso financeiro para o retorno sobre o investimento de negócios. Veja alguns exemplos:
 - Nossa prova de conceito demonstrou que podemos economizar USD X por mês em gastos com o Azure devido à otimização na nova implementação. Recomendamos implementar esse conceito na produção.
 - Os clientes têm solicitado esta nova funcionalidade, que provou ser bem-sucedida na prova de conceito. Se implementarmos esse conceito no Azure, estimamos que a receita crescerá X% por trimestre.

Se a sua prova de conceito falhou

- Você tem duas decisões:
 - Repetir a prova de conceito redefinindo a meta, o cronograma e o orçamento, o escopo e a arquitetura.
 - Realizar uma revisão post-mortem na falha para ver se há insights ou lições aprendidas durante a prova de conceito.

Nesta seção, você aprendeu a planejar e executar a prova de conceito. Na seção a seguir, vamos mostrar dois projetos práticos, completos com instruções passo a passo. Esperamos que eles inspirem você a criar seus próprios projetos de prova de conceito.



Capítulo 2: Projeto de exemplo – implementar uma aplicação Web usando Aplicativos Web Estáticos do Azure

No primeiro projeto de exemplo, você descobrirá como implementar uma aplicação Web usando Aplicativos Web Estáticos do Azure. Você aprenderá sobre o Serviço de Aplicativo do Azure no Capítulo 4: Uma visão geral do Azure para desenvolvedores, mas este projeto abordará uma das opções de hospedagem expandida do Serviço de Aplicativo: Aplicativos Web Estáticos. Os desenvolvedores podem usar Aplicativos Web Estáticos para renderizar conteúdos estáticos (como HTML, CSS e JavaScript) e, ao mesmo tempo, fornecer a lógica dinâmica necessária ao desenvolver APIs sem servidor com o Azure Functions.

O fluxo de trabalho de Aplicativos Web Estáticos (consulte Figura 3: Fluxo de trabalho de Aplicativos Web Estáticos do Azure) se assemelha ao fluxo de trabalho diário de um desenvolvedor. Aplicativos Web Estáticos fornecem um pipeline de Integração Contínua e Entrega Contínua (CI/CD) gerenciado que cria e implanta automaticamente aplicações Web de pilha completa de um repositório do GitHub no Azure. Isso é possível graças ao GitHub Actions (para saber mais sobre o GitHub Actions, acesse [esta documentação](#)).

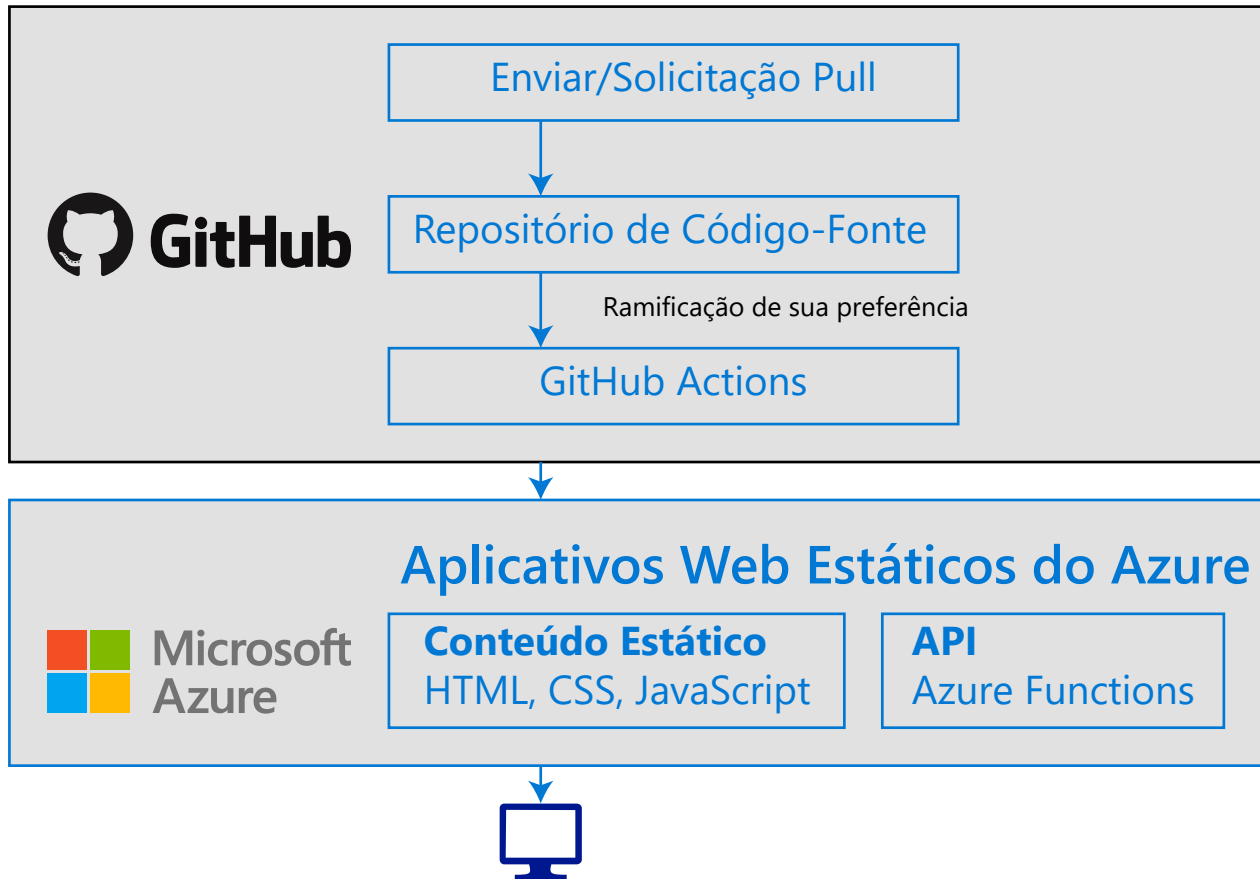


Figura 3: Fluxo de trabalho de Aplicativos Web Estáticos do Azure

Ao criar um recurso de Aplicativos Web Estáticos, o Azure configura um fluxo de trabalho do GitHub Actions no repositório de código-fonte da aplicação, que monitora uma ramificação de sua preferência. Sempre que você envia confirmações ou aceita solicitações pull na ramificação monitorada, o GitHub Actions cria e implanta automaticamente a aplicação e sua API no Azure.

Aplicativos Web Estáticos do Azure são comumente criados usando bibliotecas e estruturas como Angular, React, Svelte ou Vue. Esses aplicativos incluem HTML, CSS, JavaScript e ativos de imagem que compõem a aplicação.

Aplicativos Web Estáticos do Azure vs. servidor Web tradicional

Com um servidor Web tradicional, os ativos são fornecidos de um único servidor, juntamente com os pontos de extremidade de API obrigatórios.

Com Aplicativos Web Estáticos, os ativos estáticos são separados de um servidor Web tradicional e, em vez disso, são fornecidos de pontos distribuídos geograficamente em todo o mundo. Essa distribuição torna o fornecimento de arquivos muito mais rápido, pois os arquivos estão fisicamente mais próximos dos usuários finais.

Além disso, os pontos de extremidade de API são hospedados usando uma arquitetura sem servidor, o que reduz a necessidade de um servidor back-end completo.

Principais recursos de Aplicativos Web Estáticos do Azure

- Hospedagem na Web para conteúdo estático, como HTML, CSS, JavaScript e imagens
- Suporte integrado a APIs fornecido pelo Azure Functions
- Integração interna do GitHub em que alterações no repositório acionam builds e implantações
- Criação automática de ambientes de preparo para testar as atualizações de código antes de distribuí-las para produção
- Conteúdo estático distribuído globalmente, aproximando o conteúdo dos seus usuários
- Certificados SSL gratuitos, que são renovados automaticamente
- Domínios personalizados para fornecer personalizações de marca à sua aplicação
- Modelo de segurança contínuo com um proxy reverso ao chamar APIs, o que não requer nenhuma configuração CORS
- Integrações de provedores de autenticação com Azure Active Directory, Facebook, Google, GitHub e Twitter
- Definição e atribuições de função de autorização personalizáveis
- Regras de roteamento de back-end que proporcionam controle total sobre o conteúdo e as rotas que você atende

Caso de uso de Aplicativos Web Estáticos do Azure

- Criar aplicações de página única com estruturas e bibliotecas como Angular, React, Svelte ou Vue com um back-end do Azure Functions
- Publicar sites estáticos com estruturas como Gatsby, Hugo e VuePress
- Implantar aplicações Web com estruturas como Next.js e Nuxt.js

Colocar em prática

Observação: no momento da redação deste documento, o Aplicativo Web Estático do Azure está em visualização e é gratuito. Isso estará sujeito a alterações quando ele estiver fora da visualização.

1. Entre na sua conta do GitHub e acesse [esta URL](#) para criar um novo repositório.
2. Dê um nome ao seu repositório, como `my-first-static-web-app`. Em seguida, clique no botão **Create repository from template** (Criar repositório a partir de um modelo):

Create a new repository from angular-basic

The new repository will start with the same files and folders as [staticwebdev/angular-basic](#).

Owner * / **Repository name ***

Great repository names are short and memorable. Need inspiration? How about [fictional-guide?](#)

Description (optional)

Public
Anyone on the internet can see this repository. You choose who can commit.

Private
You choose who can see and commit to this repository.

Include all branches
Copy all branches from `staticwebdev/angular-basic` and not just `master`.

Create repository from template

Figura 4: Criar seu repositório do GitHub a partir de um modelo

3. Em seguida, vá para o [portal do Azure](#).
4. Vá para o Azure Marketplace clicando no botão **Create a resource** (Criar um recurso). Digite *Static Web App* (Aplicativo Web Estático) na caixa de pesquisa.

5. Clique em **Static Web App (preview)** (Aplicativo Web Estático [visualização]) e, em seguida, clique no botão **Create** (Criar):

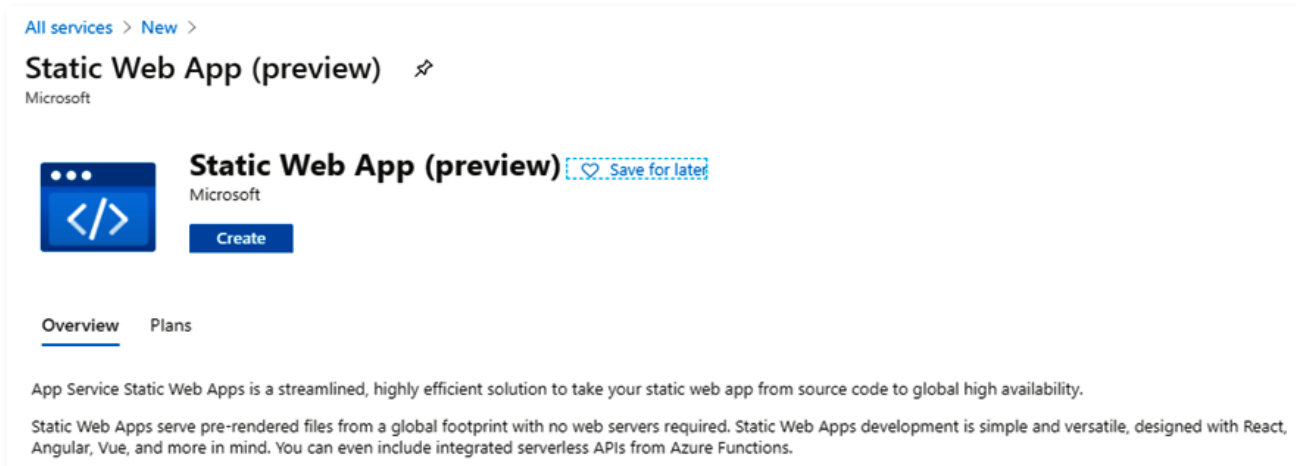


Figura 5: Criar Aplicativo Web Estático

6. Preencha o formulário:

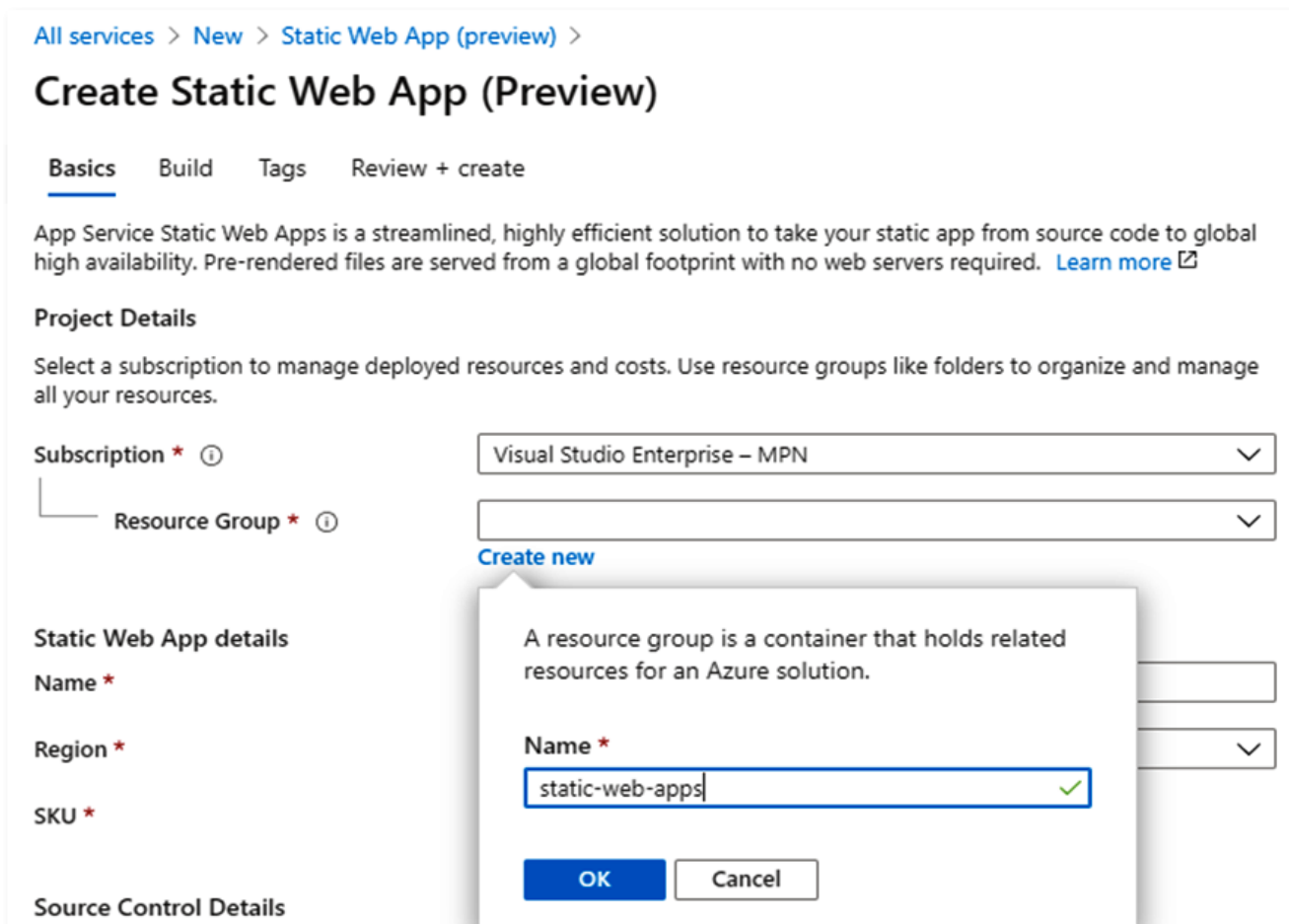


Figura 6: Criar um novo Grupo de Recursos

- Escolha uma região onde você deseja que seu Aplicativo Web Estático do Azure seja hospedado:

[All services](#) > [New](#) > [Static Web App \(preview\)](#) >

Create Static Web App (Preview)

Basics Build Tags Review + create

App Service Static Web Apps is a streamlined, highly efficient solution to take your static app from source code to global high availability. Pre-rendered files are served from a global footprint with no web servers required. [Learn more](#)

Project Details

Select a subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription * ⓘ

Resource Group * ⓘ

[Create new](#)

Static Web App details

Name *

Region *

SKU *

Source Control Details

GitHub account

Central US

East US 2

East Asia

West Europe

West US 2

Review + create

< Previous

Next : Build >

Figura 7: Escolher uma região

8. Conecte-se à sua conta do GitHub:

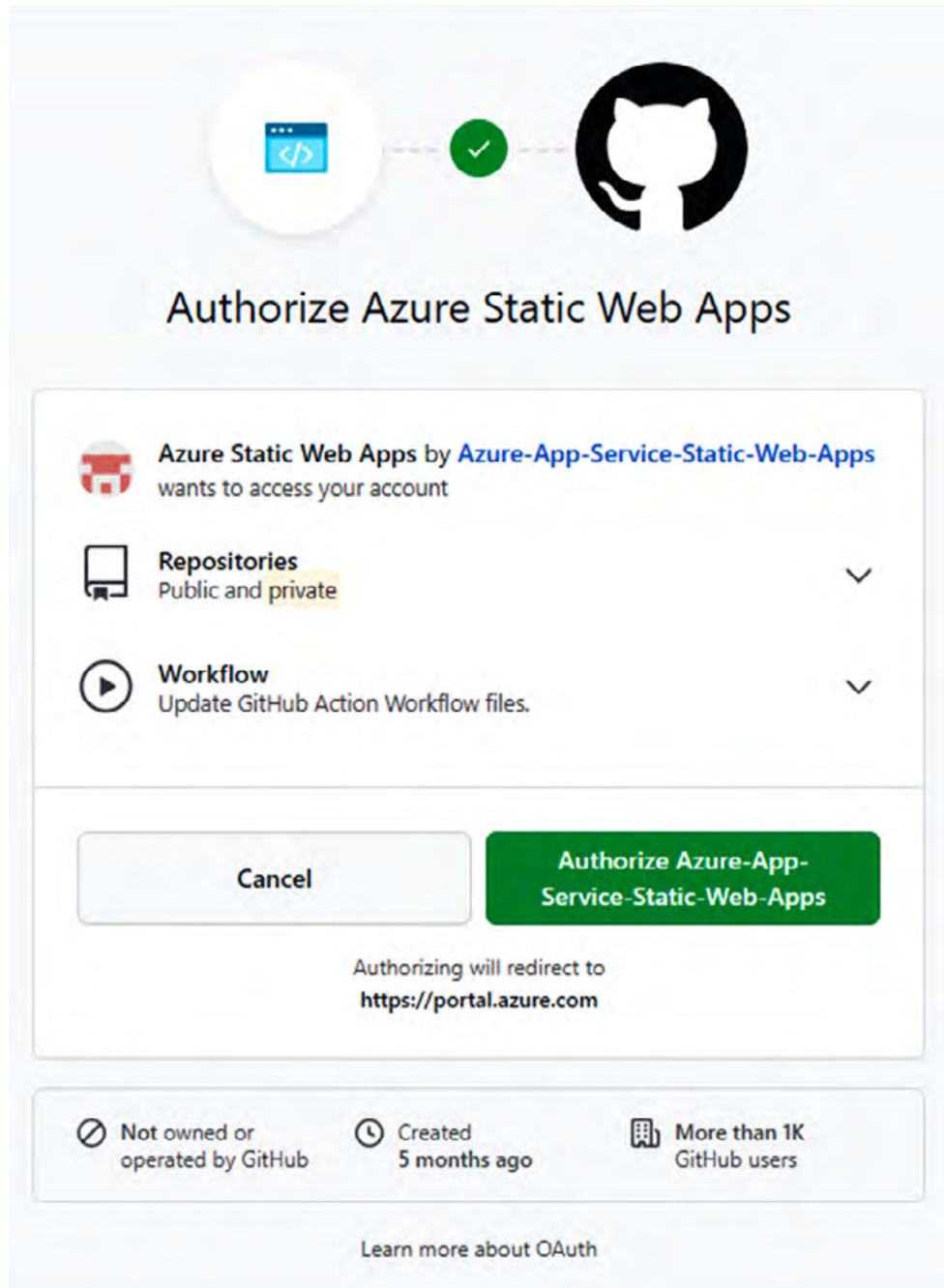


Figura 8: Conectar-se à sua conta do GitHub

9. Na seção GitHub account (Conta do GitHub), forneça as informações **Organization** (Organização), **Repository** (Repositório) e **Branch** (Ramificação) conforme mostrado:

All services > New > Static Web App (preview) >

Create Static Web App (Preview)

Basics Build Tags Review + create

App Service Static Web Apps is a streamlined, highly efficient solution to take your static app from source code to global high availability. Pre-rendered files are served from a global footprint with no web servers required. [Learn more](#)

Project Details

Select a subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription * ⓘ Visual Studio Enterprise – MPN

Resource Group * ⓘ (New) static-web-apps [Create new](#)

Static Web App details

Name * my-first-static-web-app ✓

Region * East US 2

SKU * Free

Source Control Details

GitHub account

If you can't find an organization or repository, you might need to enable additional permissions on GitHub.

Organization *

Repository * my-first-static-web-app

Branch * master

[Review + create](#) [< Previous](#) [Next : Build >](#)

Figura 9: Fornecer as informações Organization, Repository e Branch

10. Forneça as variáveis de build iniciais, conforme mostrado:

[All services](#) > [New](#) > [Static Web App \(preview\)](#) >

Create Static Web App (Preview)

Basics **Build** Tags Review + create

Provide initial build variables. These can later be modified in the workflow file.

Build Details

App location * ⓘ

Api location ⓘ

App artifact location ⓘ

Review + create

< Previous

Next : Tags >

Figura 10: Fornecer as variáveis de build iniciais

11. Navegue até a guia **Review + create** (Revisar + criar) e clique no botão **Create**:

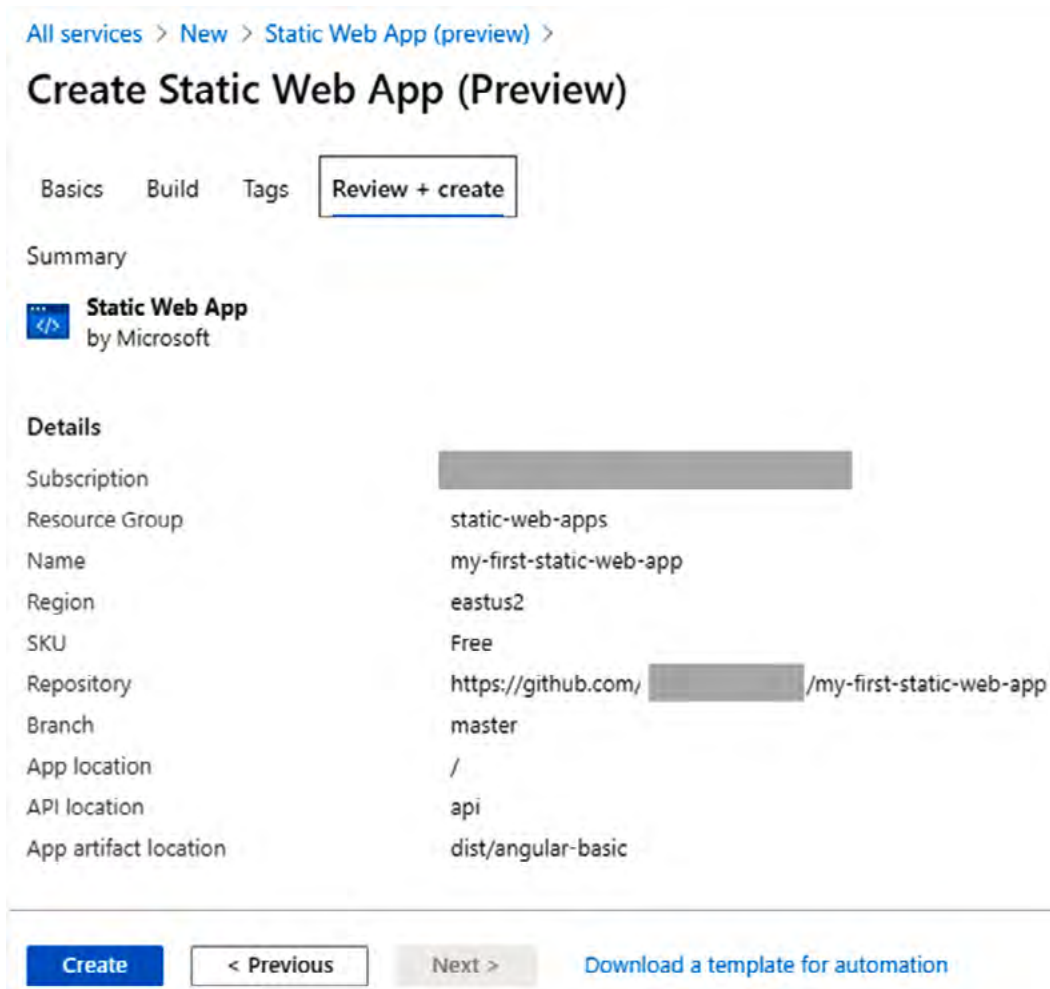


Figura 11: Review + create

12. Após a conclusão da implantação, clique no botão **Go to resources** (Ir para recursos) para encontrar a nova URL que foi gerada para seu aplicativo Web estático:

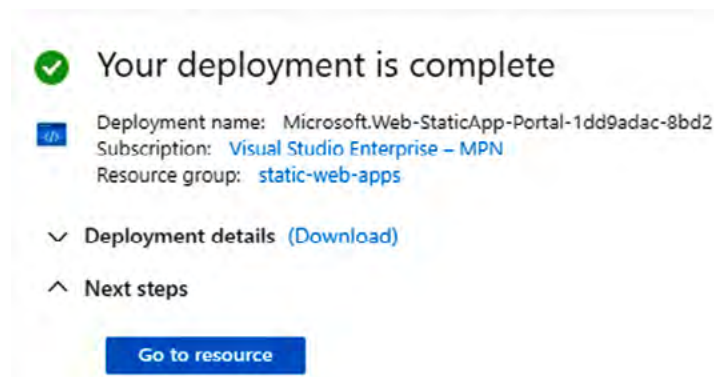


Figura 12: Implantação concluída

13. Encontre a nova URL que foi gerada para o seu aplicativo Web estático:

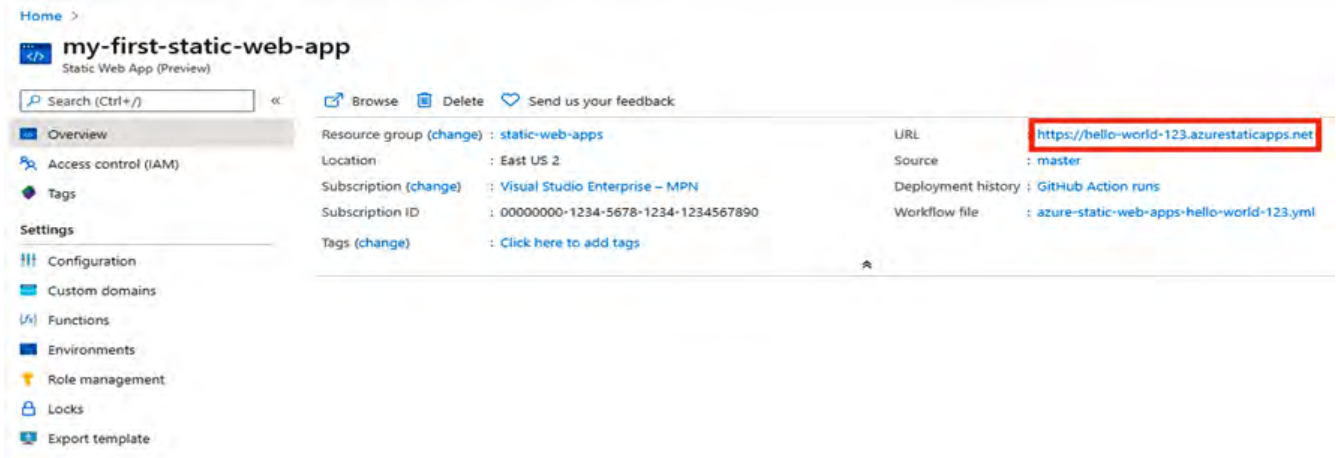


Figura 13: URL do Aplicativo Web Estático

14. Abra o navegador e acesse a URL gerada para ver seu aplicativo Web estático em ação:

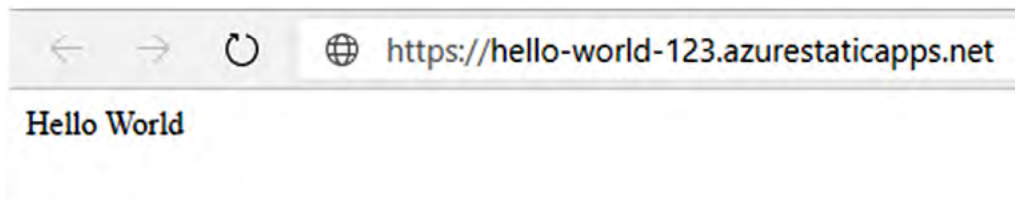


Figura 14: Aplicativo Web Estático em ação

Parabéns, você implantou com êxito seu primeiro Aplicativo Web Estático do Azure.

Para saber mais

- [Documentação oficial do GitHub Actions](#)
- [Analisar solicitações pull em ambientes de pré-produção em Aplicativos Web Estáticos do Azure](#)

A seguir, vamos explorar um segundo projeto de exemplo, criar um chatbot inteligente.



Capítulo 3: Projeto de exemplo – criar um chatbot inteligente

Neste projeto de exemplo, você explorará como criar um chatbot inteligente que use inteligência artificial (IA). Conforme a IA continua a dominar a tecnologia predominante, agora é hora de desenvolvedores como você aproveitarem o poder dela em suas aplicações.

Hoje, muitos de nós usam uma variedade de tecnologias para se comunicar. Por exemplo:

- Chamadas telefônicas
- Serviços de mensagens
- Aplicações de chat online
- Email
- Plataformas de redes sociais
- Ferramentas de colaboração

Acostumamos-nos à conectividade onipresente, e esperamos que as organizações com as quais lidamos sejam contatáveis com facilidade e imediatamente responsivas por meio dos canais que já usamos. Além disso, esperamos que essas organizações interajam conosco individualmente e sejam capazes de responder a perguntas complexas em um nível pessoal.

Principais recursos dos chatbots

Embora muitas organizações publiquem informações de suporte e respostas a perguntas frequentes (FAQs) que podem ser acessadas por meio de um navegador da Web ou uma aplicação dedicada, é difícil encontrar respostas a perguntas específicas. Essas organizações frequentemente percebem que sua equipe de suporte está sobrecarregada com solicitações de ajuda por meio de vários canais, incluindo chamadas telefônicas, email, mensagens de texto e redes sociais.

Muitas empresas estão progressivamente recorrendo a soluções de IA que fazem uso de agentes de IA (conhecidos como chatbots) para fornecer uma primeira linha de suporte automatizado por meio de toda a gama de canais que usamos para nos comunicar. Os bots foram criados para interagir com os usuários em forma de conversa, conforme mostrado na Figura 15: Um exemplo de interface de usuário de chatbot:

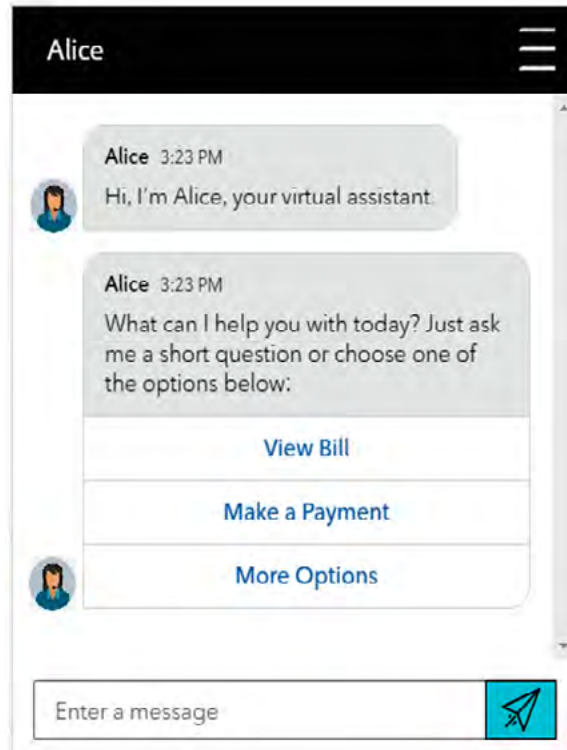


Figura 15: Um exemplo de interface de usuário de chatbot

O exemplo mostrado aqui é uma interface de chatbot como as que você encontra em sites de lojas de varejo. No entanto, os bots podem ser projetados para funcionar em vários canais, incluindo email, plataformas de redes sociais e até mesmo chamadas de voz. Independentemente do canal usado, os chatbots costumam gerenciar fluxos de conversas usando uma combinação de linguagem natural e respostas com opções restritas que orientam o usuário para uma resolução.

Caso de uso dos chatbots

A Figura 16: Azure QnA Maker e Serviço de Bot do Azure ilustra os dois componentes principais na criação de um chatbot inteligente. O primeiro componente é uma base de dados de conhecimento de perguntas e respostas. No Azure, é o QnA Maker. O segundo componente é um serviço de bot que fornece uma interface à base de dados de conhecimento. No Azure, é o Serviço de Bot do Azure. Usando o QnA Maker e o Serviço de Bot do Azure, você pode criar um chatbot que responda às perguntas frequentes dos usuários. A interface do bot pode ser uma seção de chat no seu site.

Normalmente, as conversas assumem a forma de mensagens trocadas de maneira alternada. Um dos tipos mais comuns de conversa é uma pergunta seguida por uma resposta. Esse padrão constitui a base para muitos bots de suporte ao usuário e muitas vezes pode ser baseado na documentação de perguntas frequentes existente.

Os dois componentes principais na criação da prova de conceito de um chatbot inteligente

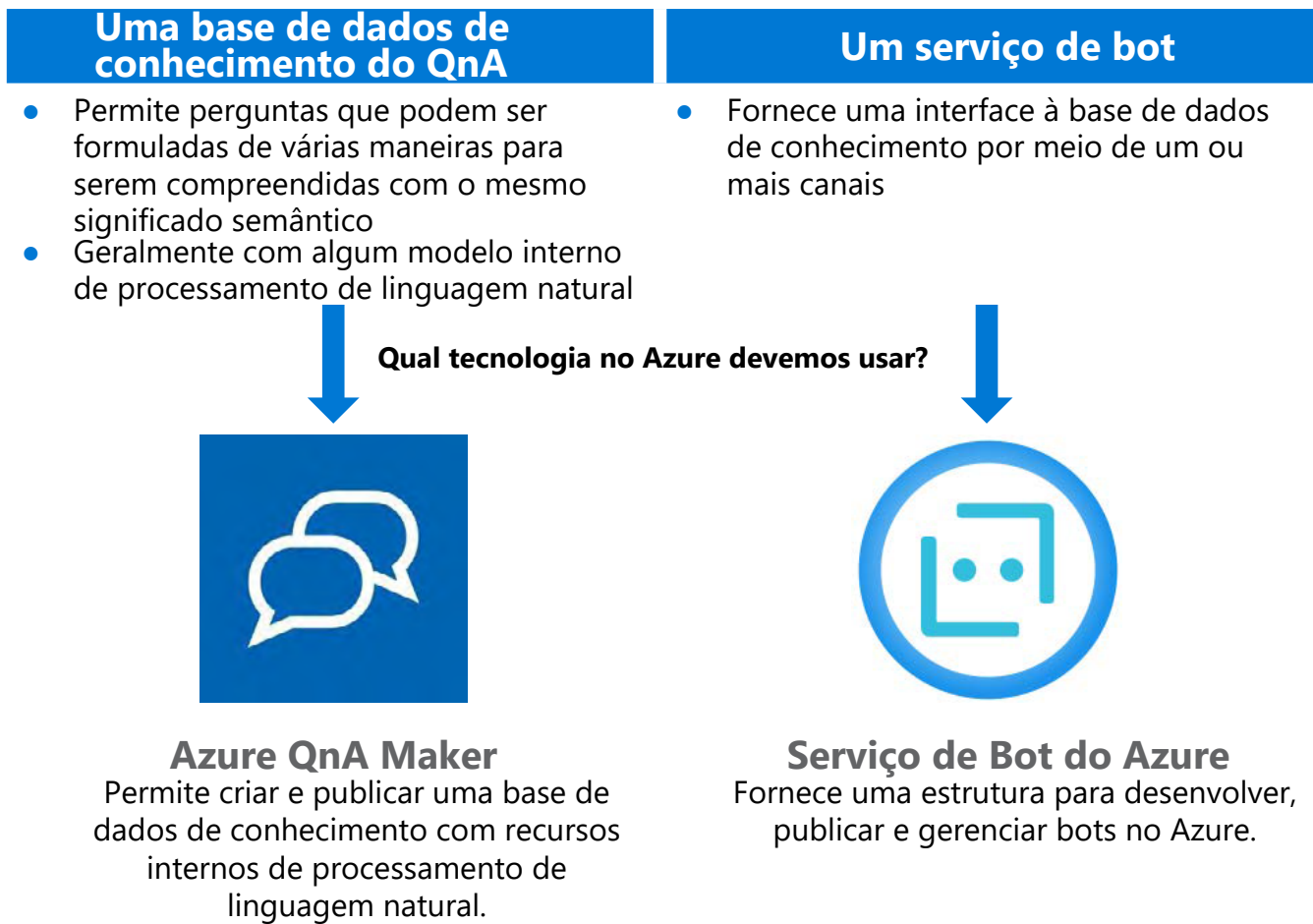


Figura 16: Azure QnA Maker e Serviço de Bot do Azure

Colocar em prática

1. Entre no [portal do QnA Maker](#) usando suas credenciais do Azure (se você ainda não tiver uma assinatura do Azure, veja a seção de assinatura do Azure no Capítulo 4: Uma visão geral do Azure para desenvolvedores).
2. Clique em **Create a knowledge base** (Criar uma base de dados de conhecimento) e, em seguida, no botão **Create a QnA service** (Criar um serviço QnA) para criar um novo recurso do QnA Maker:

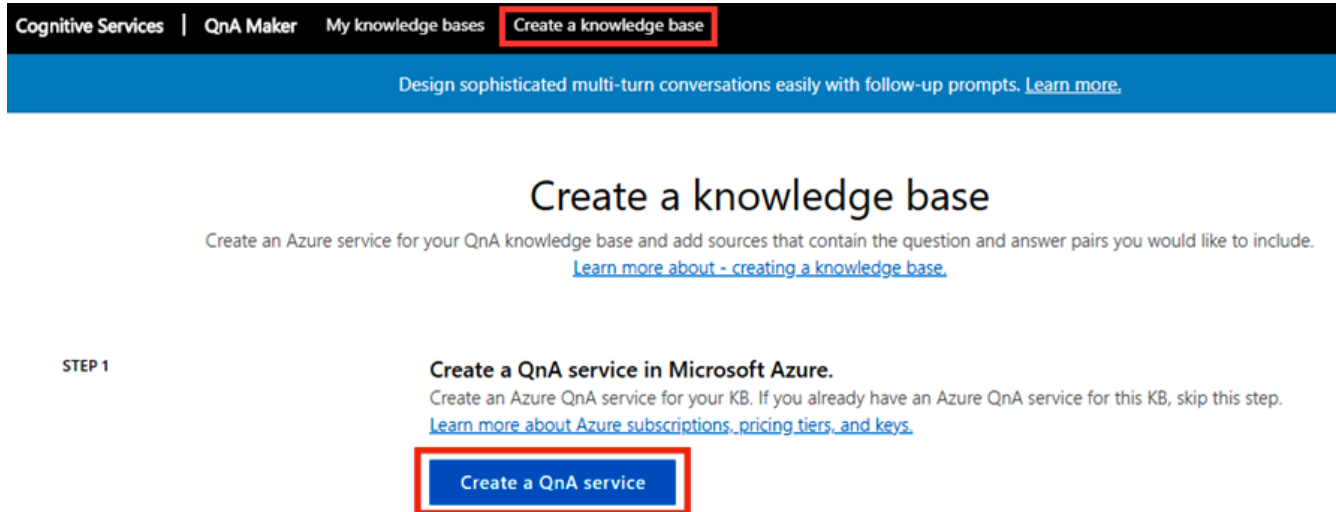


Figura 17: Criar um novo recurso do QnA Maker

- Depois que o portal do Azure for iniciado, preencha o formulário da seguinte maneira e, em seguida, clique na guia **Review + create**. A entrada no campo **Name** (Nome) deve ser globalmente exclusiva. Se você receber um erro de conflito de nome, basta tentar novamente com outro nome exclusivo:

Home >

Create

QnA Maker

*** Basics** Tags Review + create

QnA Maker is a cloud-based API service that lets you create a conversational question-and-answer layer over your existing data. Use it to build a knowledge base by extracting questions and answers from your semi-structured content, including FAQs, manuals, and documents. Answer users' questions with the best answers from the QnAs in your knowledge base—automatically. Your knowledge base gets smarter, too, as it continually learns from user behavior. [Learn more](#)

Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription * ⓘ Microsoft Azure MVP

Resource group * ⓘ (New) my-qnamaker-poc
[Create new](#)

Name * ⓘ my-qnamaker-poc

Pricing tier * ⓘ Free F0 (3 managed documents per month, 3 transactions per second...)

Azure Search details - for data

When you create a QnAMaker resource, you host the data in your own Azure subscription. Azure Search is used to index your data.

Azure Search location * (US) East US

Azure Search pricing tier * ⓘ Free F (3 Indexes)

App Service details - for runtime

When you create a QnAMaker resource, you host the runtime in your own Azure subscription. App Service is the compute engine that runs the QnA Maker queries for you.

App name * ⓘ my-qnamaker-poc

Website location * (US) East US

Review + create Next : Tags >

Figura 18: Adicionar detalhes na guia Basics (Informações básicas)

- Depois que seu recurso do QnA Maker for provisionado, volte para [Create a knowledge base](#), atualize a página e prossiga para a Etapa 2:

STEP 2

Connect your QnA service to your KB.

After you create an Azure QnA service, refresh this page and then select your Azure service using the options below

Refresh

* Microsoft Azure Directory ID

* Azure subscription name

* Azure QnA service

* Language

Figura 19: Etapa 2 do QnA Maker – Conectar o serviço QnA à sua base de dados de conhecimento

- Na Etapa 3, nomeie sua base de dados de conhecimento:

STEP 3

Name your KB.

The knowledge base name is for your reference and you can change it at anytime.

* Name

Figura 20: Etapa 3 do QnA Maker – Nomear sua base de dados de conhecimento

6. Preencha a Etapa 4 da seguinte maneira:

STEP 4

Populate your KB.

Extract question-and-answer pairs from an online FAQ, product manuals, or other files. Supported formats are .tsv, .pdf, .doc, .docx, .xlsx, containing questions and answers in sequence. [Learn more about knowledge base sources](#). Skip this step to add questions and answers manually after creation. The number of sources and file size you can add depends on the QnA service SKU you choose. [Learn more about QnA Maker SKUs](#).

Enable multi-turn extraction from URLs, .pdf or .docx files. [Learn more](#).

* Default answer text ?

Sorry, I don't have an answer for you.

URL

<https://docs.microsoft.com/azure/cognitive-services/qnamaker/troubleshooting>

+ Add URL

File name

+ Add file

Chit-chat

Give your bot the ability to answer thousands of small-talk questions in a voice that fits your brand. When you add chit-chat to your knowledge base by selecting a personality below, the questions and responses will be automatically added to your knowledge base, and you'll be able to edit them anytime you want. [Learn more about chit-chat](#).

None

Professional

Friendly

Witty

Caring

Enthusiastic

Figura 21: Etapa 4 do QnA Maker – Preencher sua base de dados de conhecimento

7. Clique no botão **Create your KB** (Criar sua base de dados de conhecimento) na Etapa 5:

STEP 5

Create your KB

The tool will look through your documents and create a knowledge base for your service. If you are not using an existing document, the tool will create an empty knowledge base table which you can edit.

Create your KB

Figura 22: Etapa 5 do QnA Maker – Criar sua base de dados de conhecimento

8. No portal do QnA Maker, na página **Edit** (Editar), selecione + **Add QnA pair** (+ Adicionar par de pergunta e resposta) na barra de ferramentas:

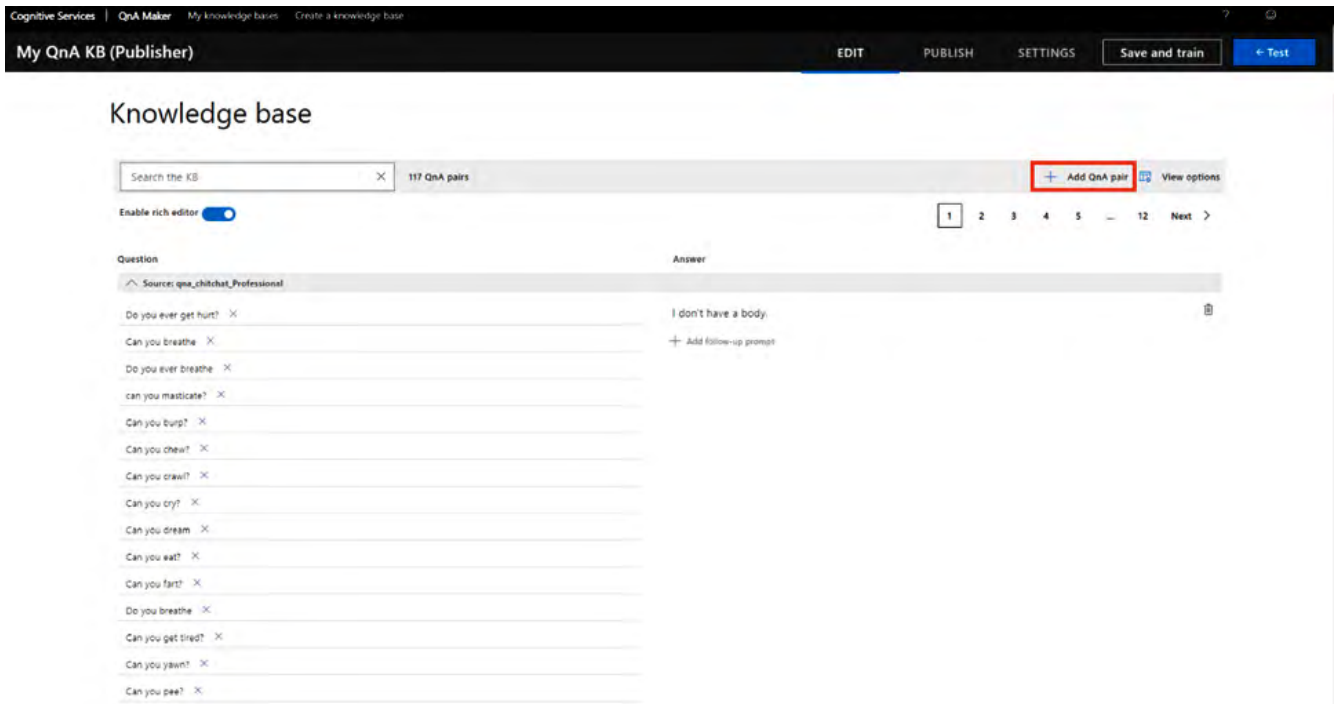


Figura 23: Adicionar um par de pergunta e resposta

9. Adicione a pergunta e a resposta. Em seguida, clique no botão **Save and train** (Salvar e treinar):

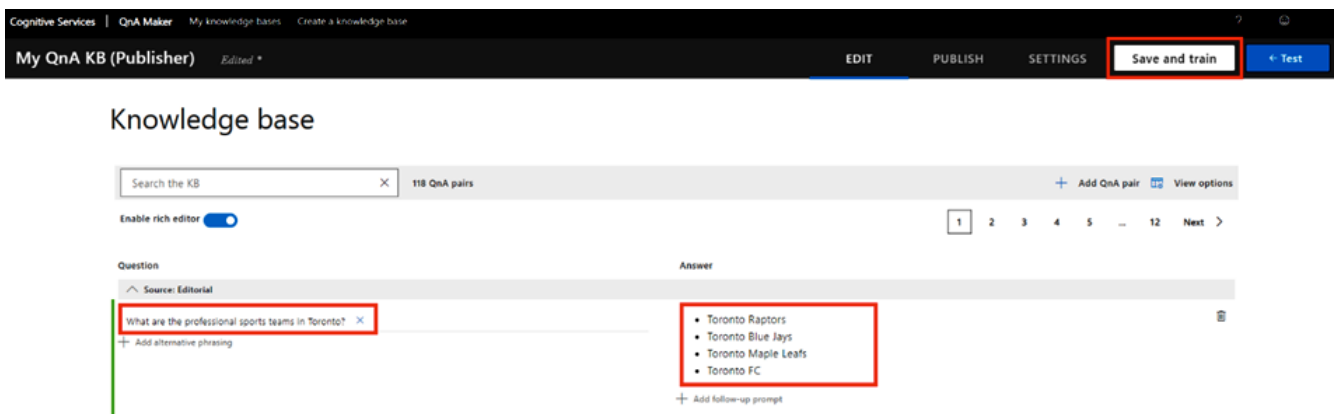


Figura 24: Adicionar perguntas e respostas

10. Você pode testar sua base de dados de conhecimento imediatamente pressionando o botão **Test** (Testar). Em seguida, insira uma pergunta como *“What are the professional sports teams in Toronto”* (Quais são as equipes esportivas profissionais em Toronto):

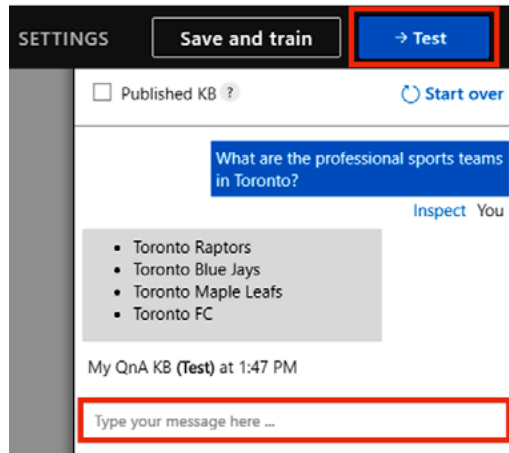


Figura 25: Testar sua base de dados de conhecimento

11. Pressione **Inspect** (Inspeccionar) para examinar os detalhes da conversa:

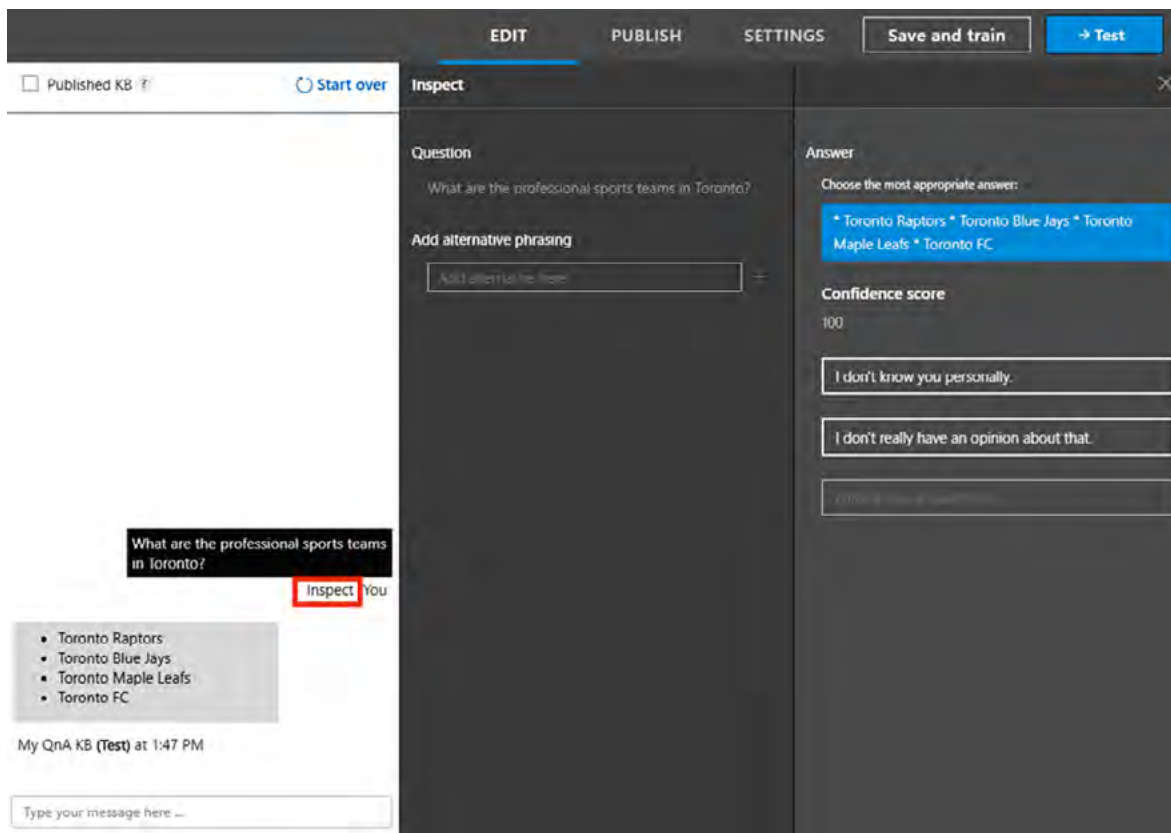


Figura 26: Examinar os detalhes da conversa

12. Clique no botão **Test** para fechar o painel de teste.

Publicar sua base de dados de conhecimento do QnA

Quando você publica sua base de dados de conhecimento, o conteúdo muda do índice de teste para um índice de produção na Pesquisa do Azure.

No portal do QnA Maker, clique no botão **Publish** (Publicar):

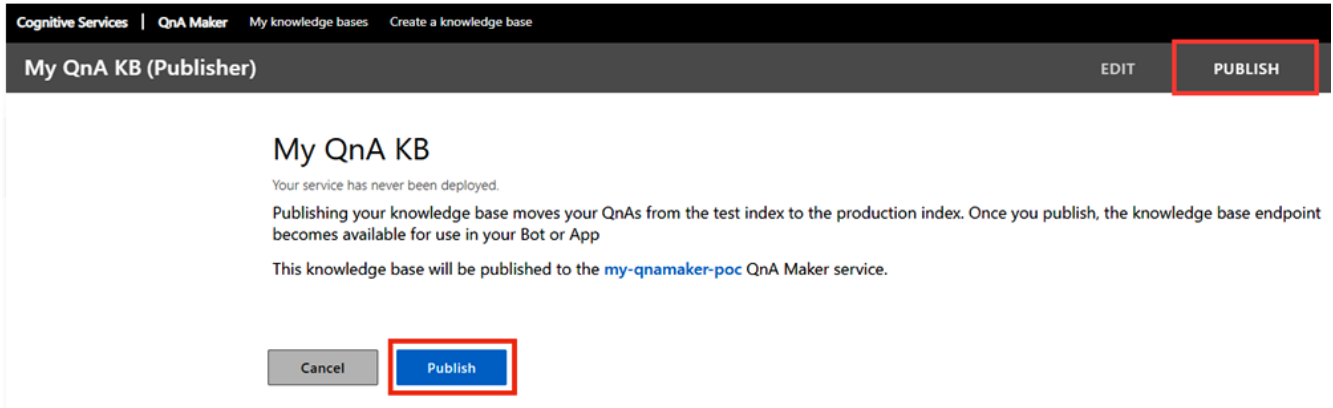


Figura 27: Publicar sua base de dados de conhecimento do QnA

Criar o bot no Serviço de Bot do Azure

Em seguida, você criará um bot no Serviço de Bot do Azure para vincular à base de dados de conhecimento que criou nas etapas anteriores:

1. Após a implantação bem-sucedida da sua base de dados de conhecimento, clique no botão **Create Bot** (Criar Bot) para iniciar a página de criação do Serviço de Bot do Azure no portal do Azure:

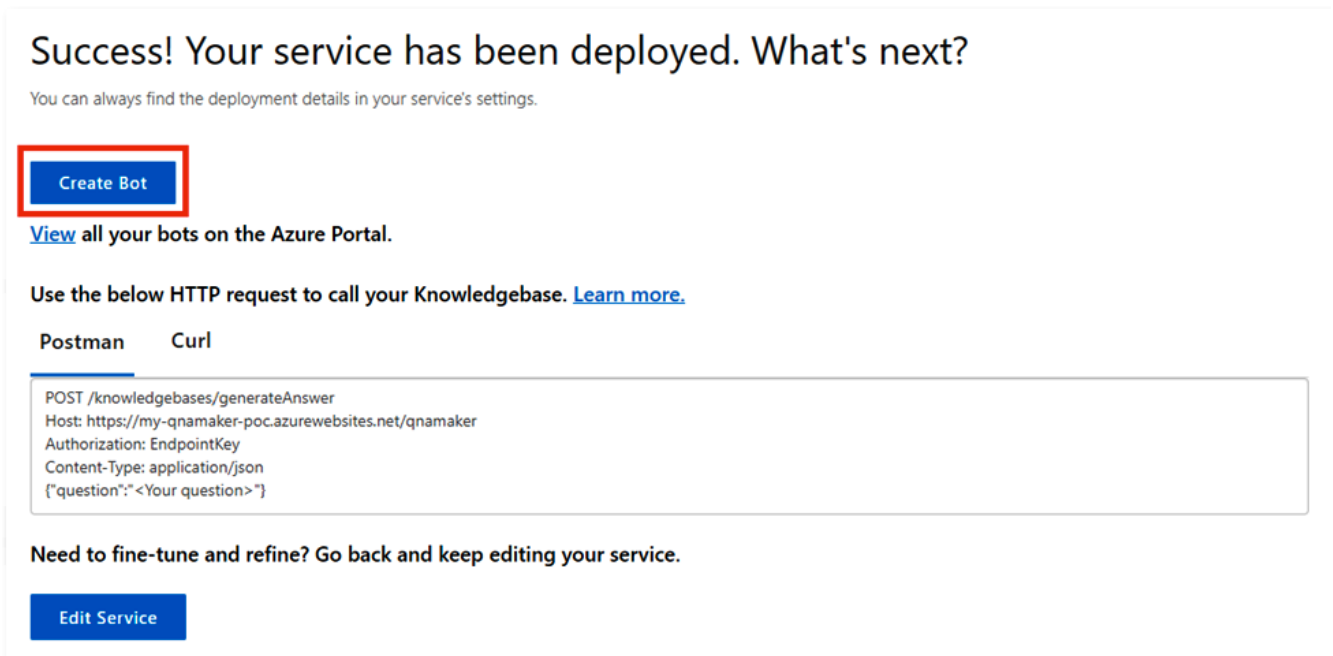


Figura 28: Iniciar a página de criação do Serviço de Bot do Azure

- Na página de criação do Serviço de Bot do Azure, preencha o formulário da seguinte maneira:

Home >

Web App Bot

Bot Service

Bot handle * ⓘ

Subscription *

Resource group *
 [Create new](#)

Location * ⓘ

Pricing tier (View full pricing details)

App name * ⓘ
 [.azurewebsites.net](#)

SDK language *
 C# Node.js

QnA Auth Key * ⓘ

***App service plan/Location** >
 my-qnamaker-poc/East US

Application Insights ⓘ
 On Off

Microsoft App ID and password ⓘ >
 Auto create App ID and password

Create Automation options

Figura 29: Adicionar detalhes do Bot do Aplicativo Web

3. Depois que o bot for provisionado, abra-o no Bot Services (Serviços de Bot):

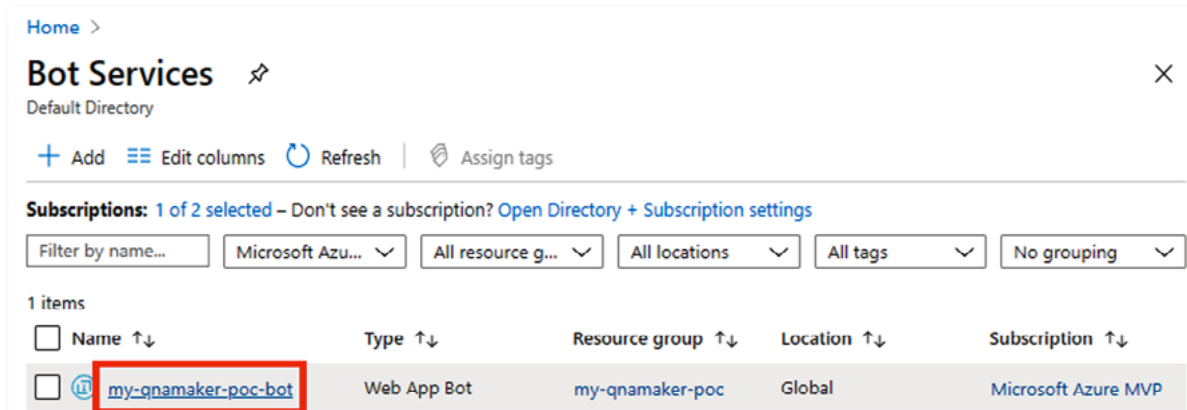


Figura 30: Abrir bot no Serviço de Bot

4. Em **Bot management** (Gerenciamento de bots), selecione **Test in Web Chat** (Testar no Web Chat):

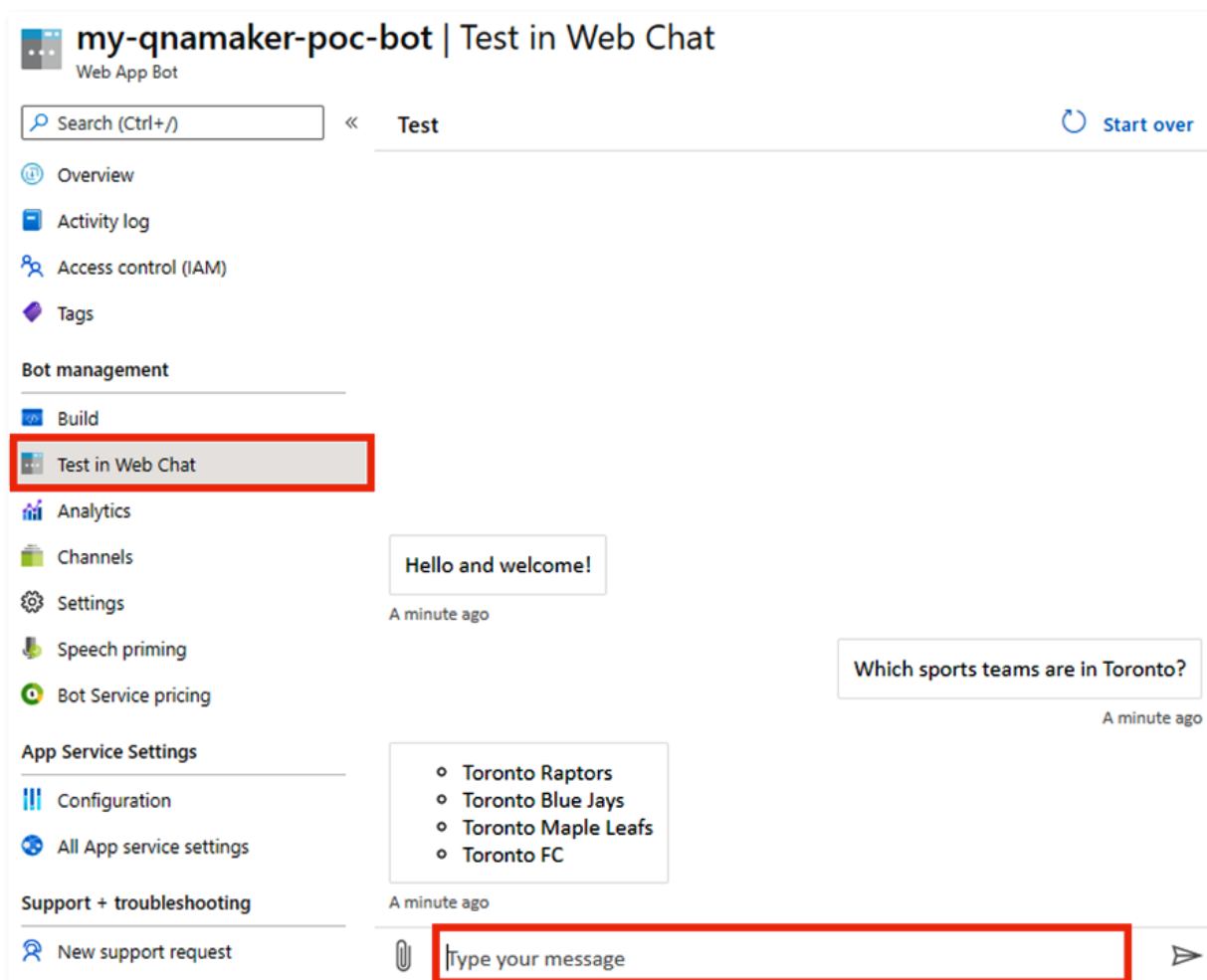


Figura 31: Selecionar Test in Web Chat

Resumindo, você usou o QnA Maker para criar uma nova base de dados de conhecimento no Azure. Você adicionou uma URL pública à base de dados de conhecimento. Em seguida, você adicionou seu próprio par de pergunta e resposta, treinou e testou. Depois de publicar sua base de dados de conhecimento, você criou um Bot do Aplicativo Web no Serviço de Bot do Azure. Por fim, você testou o bot em execução no Azure. Os bots podem ajudar a reduzir os custos de suporte fornecendo suporte automatizado por meio de vários canais de comunicação. Mostramos como usar o QnA Maker e o Serviço de Bot do Azure para criar um chatbot que responda às perguntas de suporte dos usuários.

Isso conclui nosso tour de dois projetos práticos. Esperamos que eles inspirem você a criar seus próprios projetos de prova de conceito. Na próxima seção, vamos explorar uma visão geral do Azure para desenvolvedores e aprender a escolher o modelo de nuvem apropriado para seus projetos de prova de conceito do Azure.



Capítulo 4: Uma visão geral do Azure para desenvolvedores

Introdução ao Microsoft Azure

Conforme discutido na introdução, o Azure é uma plataforma de nuvem que capacita você a:

- Hospedar suas aplicações existentes com alterações mínimas
- Desenvolver novas aplicações nativas de nuvem

Há um grande número de serviços do Azure prontos para uso que podem ser integrados às suas aplicações para aproveitar instantaneamente os novos recursos de nuvem, minimizando a necessidade de você mesmo desenvolver esses serviços. Nesta seção, vamos discutir:

- Os benefícios do Azure
- Como começar a usar o Azure
- Ferramentas necessárias para desenvolver sua aplicação para o Azure
- Como escolher o modelo de nuvem apropriado para seu projeto de prova de conceito do Azure

Os benefícios do Azure

Veja a seguir alguns dos principais benefícios de hospedar suas aplicações no Azure:

Custos mais baixos

- Ao desenvolver aplicações no Azure, sua empresa economizará em despesas de capital a longo prazo graças ao modelo de preços pré-pago do Azure.
- Você paga apenas pelos serviços que usa.
- Não é necessário realizar despesas de capital comprando servidores na infraestrutura local.

Menos manutenção

- Com os serviços gerenciados, você pode se concentrar no desenvolvimento da sua aplicação.
- O Azure cuidará da manutenção da infraestrutura para você.

Escalabilidade

- Os serviços do Azure são altamente escaláveis para atender às crescentes demandas da sua empresa.

Alta disponibilidade e confiabilidade

- Você pode configurar seus serviços do Azure para garantir que suas aplicações sejam altamente disponíveis e confiáveis.

Assinatura do Azure

Para começar a criar aplicações para serem executadas no Azure, você precisará de uma assinatura do Azure. Se a sua empresa fornecer créditos do Azure, está tudo pronto. Caso contrário, basta se inscrever para uma [conta gratuita do Azure](#).

O que a conta gratuita do Azure inclui?

A conta gratuita do Azure permite que você comece com 12 meses de serviços gratuitos e crédito de USD 200 para explorar o Azure por 30 dias. Essas ofertas podem mudar com o passar do tempo. Para obter os detalhes mais atualizados sobre o que está incluído na conta gratuita do Azure, acesse as [Perguntas frequentes sobre a conta gratuita do Azure](#).

Se você é novo no Azure, há muitos recursos gratuitos para ajudá-lo a começar com facilidade. Você pode encontrar esses recursos no Capítulo 5: Mais aprendizagem e recursos.

Na próxima seção, você examinará as ferramentas necessárias para desenvolver seu projeto de prova de conceito para o Azure.

Ferramentas necessárias para desenvolver seu projeto de prova de conceito para o Azure

Veja a seguir uma lista de ferramentas essenciais para os exemplos mostrados neste guia:

- [Assinatura do Azure](#)
- [Visual Studio Code](#)
- [Azure Functions Core Tools](#)
- [Conta do GitHub](#)
- [Navegador Microsoft Edge \(baseado no Chromium\)](#)
- [Node.js](#)

Na próxima seção, você aprenderá a escolher o modelo de nuvem apropriado para seu projeto de prova de conceito do Azure.

Como escolher o modelo de nuvem apropriado para seu projeto de prova de conceito do Azure

No Capítulo 1: Guia de prova de conceito, você aprendeu sobre a prova de conceito. Quando tiver seu planejamento de prova de conceito em vigor, você poderá iniciar a implementação e os testes. Nesta seção, mostraremos os quatro modelos de nuvem (consulte Figura 32: Modelos de nuvem do Azure para Desenvolvimento e Hospedagem de Aplicações) que estão disponíveis para você escolher para seu projeto de prova de conceito do Azure. Explicaremos o caso de uso de cada um desses modelos de nuvem para que você possa fazer uma escolha embasada.

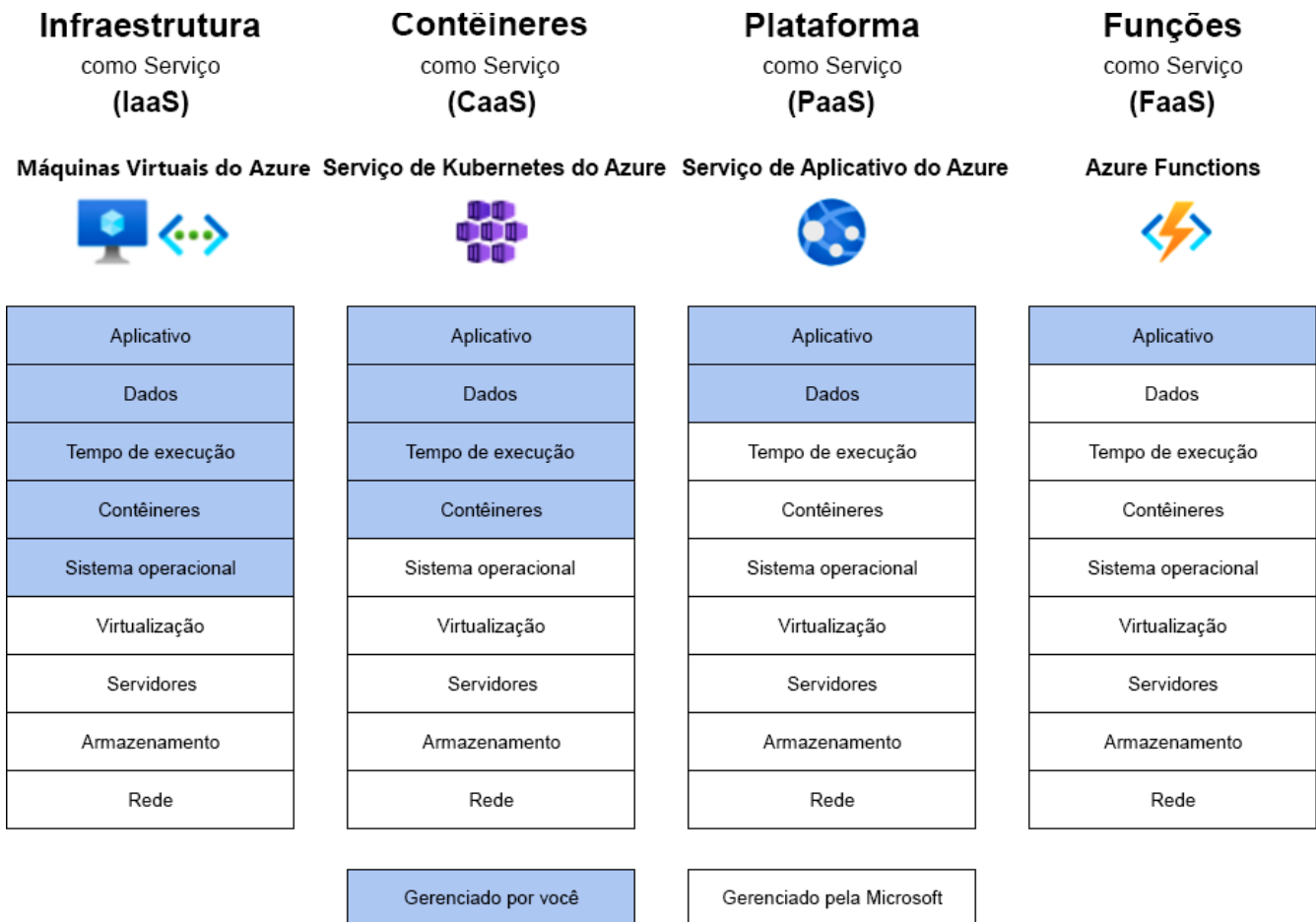


Figura 32: Modelos de nuvem do Azure para Desenvolvimento e Hospedagem de Aplicações

Infraestrutura como Serviço (IaaS)

No modelo de IaaS, o Azure oferece controle total para gerenciar seu próprio ambiente de hospedagem de aplicações.

Introdução às Máquinas Virtuais do Azure

No modelo de IaaS, o Azure permite implantar ou migrar sua aplicação para máquinas virtuais Windows ou Linux. Você terá controle total sobre a configuração da máquina. No entanto, no modelo de IaaS, você é responsável por todas as atualizações do sistema operacional, instalação do software de servidor, configuração e manutenção.

Principais recursos das Máquinas Virtuais do Azure

- As Máquinas Virtuais do Azure permitem ter controle completo do sistema operacional.
- Você pode ajustar o equilíbrio entre CPU/memória, a família de máquinas (SKU) e os layouts de disco.

Caso de uso das Máquinas Virtuais do Azure

- As Máquinas Virtuais do Azure são uma boa opção se você desejar ter controle total sobre sua infraestrutura de aplicações ou migrar workloads de aplicações na infraestrutura local para o Azure com pouca ou nenhuma alteração.

Para saber mais

- [Documentação das Máquinas Virtuais Windows](#)
- [Documentação das Máquinas Virtuais Linux](#)

Contêiner como Serviço (CaaS)

Um contêiner, por definição, é uma unidade padrão de software que agrupa um ambiente de tempo de execução inteiro (uma aplicação, arquivos de configuração necessários para executá-la e todas as suas dependências, bibliotecas e outros binários) em um único pacote. Ao organizar a plataforma de aplicações e suas dependências em contêineres, as variações nos ambientes de sistema operacional e na infraestrutura subjacente são abstraídas dela.

Introdução ao Serviço de Kubernetes do Azure

No modelo de CaaS, ao executar mais de um contêiner, é necessário ter um orquestrador. O orquestrador de contêineres gerenciado no Azure é chamado de Serviço de Kubernetes do Azure (AKS).

O AKS gerencia seu ambiente hospedado do Kubernetes e simplifica a implantação de um cluster gerenciado do Kubernetes no Azure. Você pode criar um cluster do AKS no portal do Azure, com a interface de linha de comando do Azure ou opções de implantação orientadas por modelo (como os modelos do Azure Resource Manager e o Terraform). Com o AKS, você pode implantar e gerenciar facilmente aplicações em contêineres. Não é necessário ser um especialista em orquestração de contêineres para usar o AKS.

Os nós mestres e de trabalho do Kubernetes são implantados e configurados para você ao implantar um cluster do AKS. Durante a implantação, outros recursos (como integração do Azure Active Directory, monitoramento e rede avançada) também podem ser configurados. O AKS também oferece suporte a contêineres do Windows Server.

Principais recursos do AKS

- Reduz a complexidade e a sobrecarga operacional de gerenciar o Kubernetes descarregando grande parte dessa responsabilidade para o Azure.
- O Azure cuida de tarefas críticas como monitoramento da integridade e manutenção para você.
- O Azure gerencia os mestres do Kubernetes para você.
- Os mestres do Kubernetes são gratuitos.
- Você só gerencia e mantém os nós de agente.
- Você só paga pelos nós de agente nos seus clusters.

Caso de uso do AKS

- O AKS é uma boa opção se você quiser simplificar a implantação e o gerenciamento de aplicações baseadas em microsserviços. Você também pode usar o AKS para migrar aplicações existentes para contêineres e executá-las.

Para saber mais

- [Serviço de Kubernetes do Azure](#)

Plataforma como Serviço (PaaS)

No modelo de PaaS, o Azure fornece um ambiente hospedado totalmente gerenciado para executar sua aplicação. Você não precisará se preocupar com os detalhes da infraestrutura subjacente.

Introdução ao Serviço de Aplicativo do Azure

O Serviço de Aplicativo do Azure é uma PaaS que oferece a maneira mais rápida de publicar seus projetos baseados na Web. Com o Serviço de Aplicativo, você pode estender facilmente suas aplicações Web para oferecer suporte aos clientes móveis e publicar APIs REST.

Além disso, pode criar os seguintes tipos de aplicações:

- Aplicações Web
- APIs
- Back-ends de aplicação móvel

Como todos esses tipos de aplicações compartilham o tempo de execução do Serviço de Aplicativo, você pode literalmente hospedar um site, oferecer suporte a clientes móveis e expor suas APIs no Azure de um único projeto ou solução.

O Serviço de Aplicativo foi projetado com o DevOps em mente. Ele oferece suporte a várias ferramentas para publicação e CI/CD, como:

- Azure DevOps
- GitHub
- Bitbucket
- Docker Hub
- Registro de Contêiner do Azure

Principais recursos do Serviço de Aplicativo do Azure

- Autenticação usando provedores de redes sociais (como Conta da Microsoft, Facebook, Twitter, Google)
- Dimensionamento automático baseado em tráfego
- Testes na produção
- Implantações contínuas e baseadas em contêineres

Caso de uso do Serviço de Aplicativo do Azure

- O Serviço de Aplicativo do Azure oferece suporte a uma ampla gama de tecnologias da Web
- Ótima maneira de criar prova de conceitos para quase qualquer serviço Web ou frontend da Web (como .NET, Java, Python, PHP, entre outros)
- O Serviço de Aplicativo do Azure é uma boa opção quando:
 - Você está migrando uma aplicação Web existente para o Azure.
 - Você precisa de uma plataforma de hospedagem totalmente gerenciada para suas aplicações Web.
 - Você precisa implantar e executar uma aplicação Web em contêineres.
 - Você precisa expor APIs REST com sua aplicação.

Migrar para o Serviço de Aplicativo do Azure

- A ferramenta Migrar para o Serviço de Aplicativo do Azure auxilia na migração de aplicações .NET e PHP existentes para o Serviço de Aplicativo do Azure.

Para saber mais

- [Visão geral do Serviço de Aplicativo do Azure](#)

Colocar em prática

Para colocar o Serviço de Aplicativo em ação, acesse [Try Azure App Service \(Experimentar o Serviço de Aplicativo do Azure\)](#).

Isso permite que você provisione uma aplicação de prova de conceito de curto prazo e experimente a plataforma em um ambiente de área restrita sem a necessidade de uma assinatura do Azure. É gratuito e não há compromisso.

1. Na página principal, você pode selecionar **Web App** (Aplicação Web) ou **Web App for Containers** (Aplicação Web para Contêineres). Vamos escolher **Web App**:

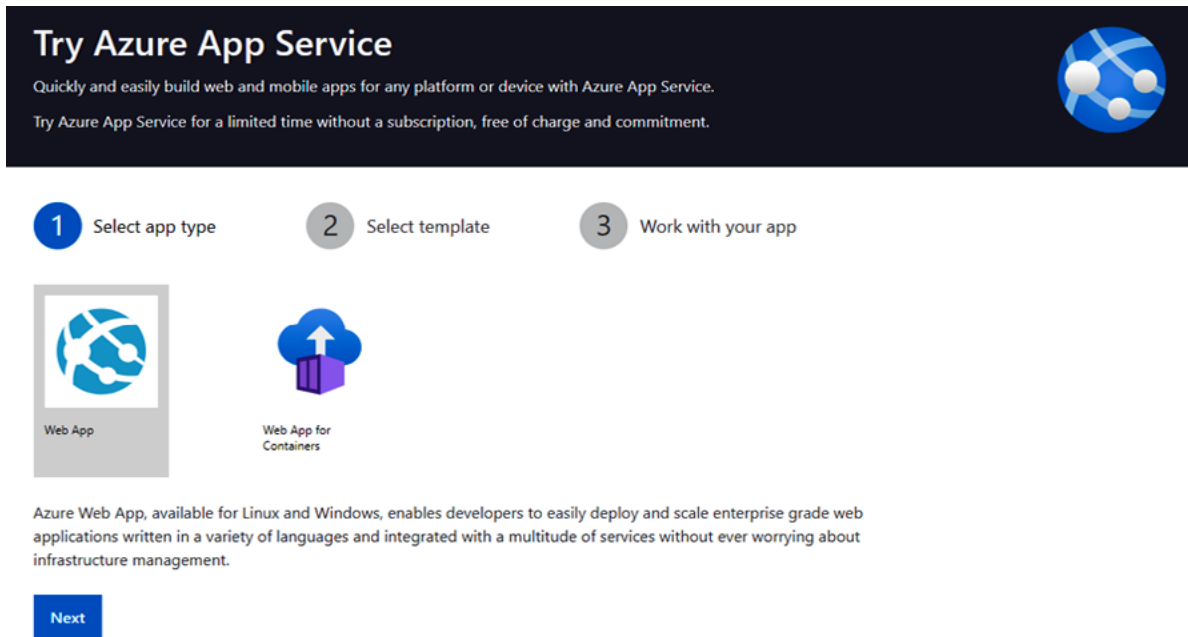


Figura 33: Escolher Web App ou Web App for Containers

2. Em seguida, escolha uma linguagem para selecionar um modelo para criar sua Aplicação Web. Vamos escolher C# e ASP.NET Core. Em seguida, clique no botão **Create**:

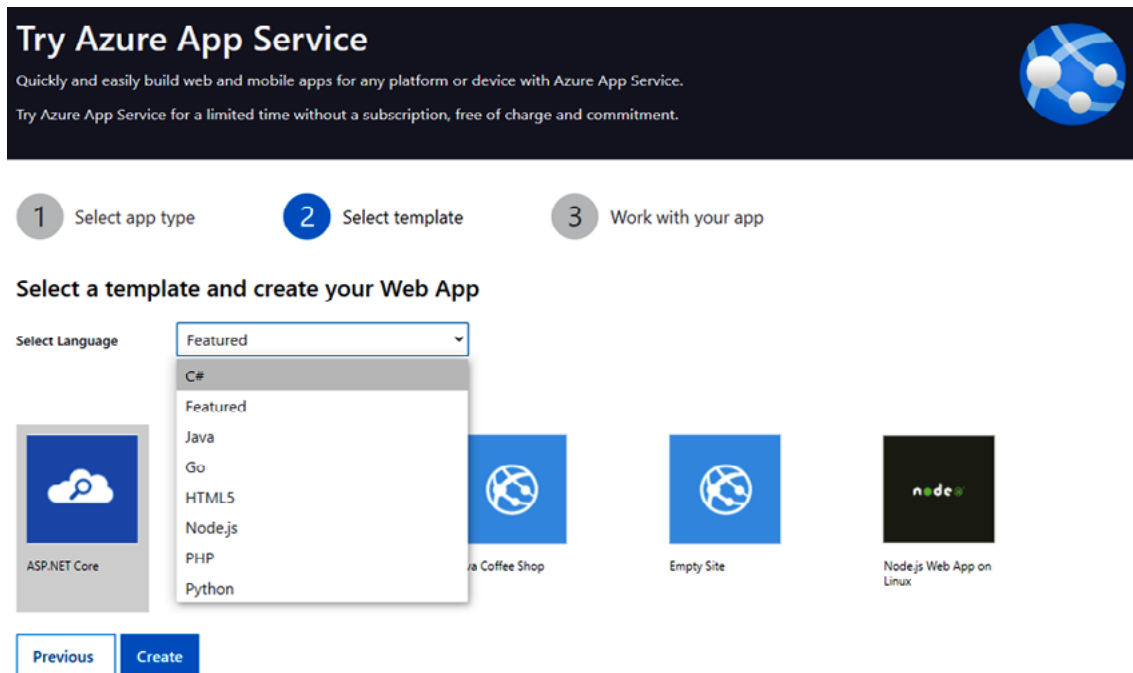


Figura 34: Escolher uma linguagem

3. Na próxima etapa, você precisará entrar para criar sua Aplicação Web usando qualquer uma destas quatro opções:



Figura 35: Entrar para criar sua Aplicação Web

4. Depois que isso for implantado com êxito, você poderá fazer alterações no conteúdo e experimentar os resultados alterados:

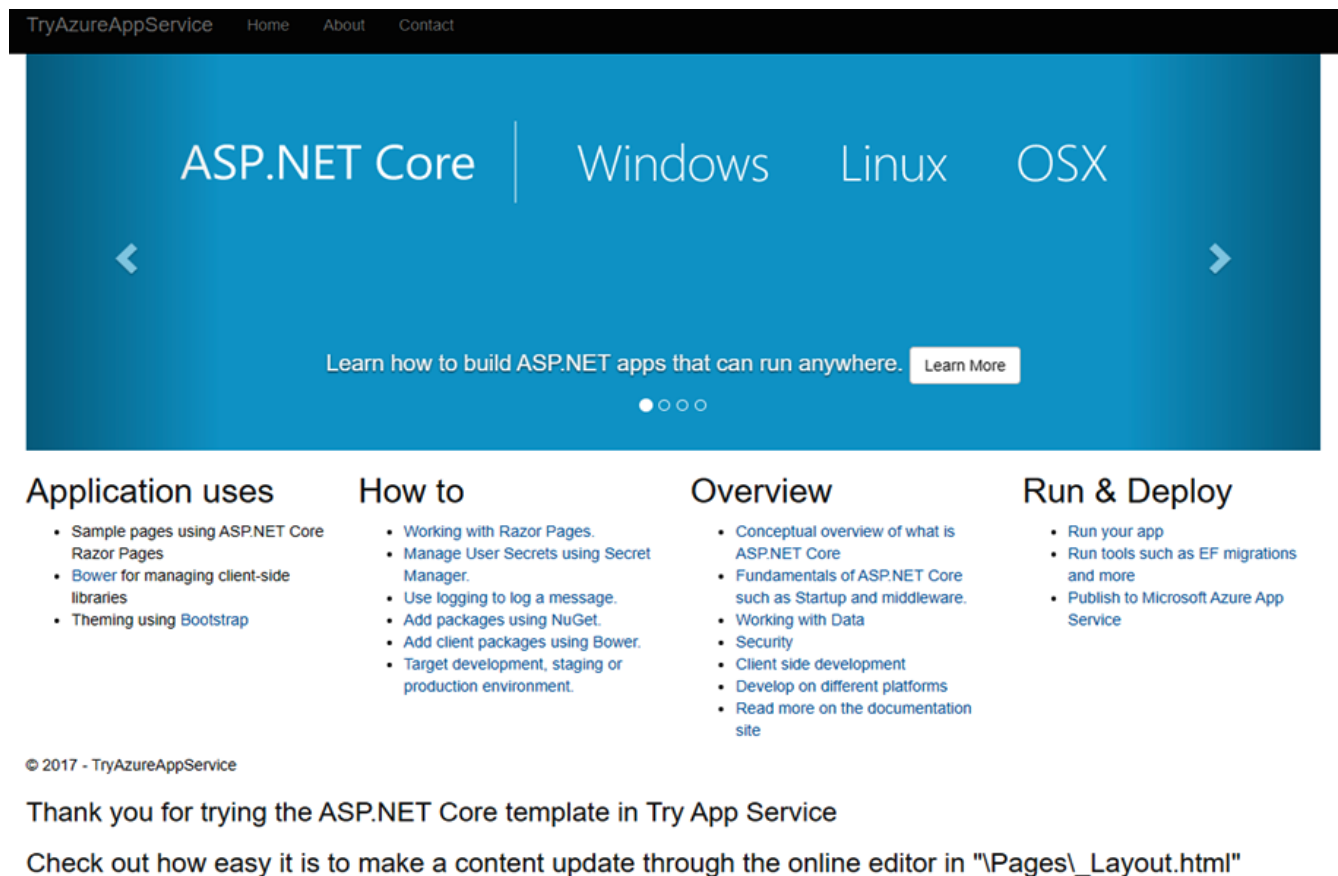


Figura 36: Site implantado com êxito

5. Quando terminar com esta aplicação de prova de conceito, volte para o site [Try Azure App Service](#) e experimente outro modelo.

A seguir, vamos considerar a oferta sem servidor conhecida como Azure Functions.

Função como Serviço (FaaS)

Com o modelo de FaaS, o Azure fornece um ambiente sem servidor. Tudo o que você precisa fazer é desenvolver seu código. Os detalhes da infraestrutura subjacente são todos gerenciados pelo Azure.

Introdução ao Azure Functions

O Azure Functions permite executar código sem servidor sem a necessidade de provisionar sua própria infraestrutura. Uma função do Azure é uma unidade lógica de código que pode ser acionada por solicitações HTTP, um evento em outro serviço do Azure ou em uma agenda.

O Azure Functions é sem servidor porque você pode se concentrar em escrever seu código sem ter que se preocupar com um servidor que executa o código. Você só é cobrado quando o ponto de extremidade é chamado. Quando os pontos de extremidade não estão sendo usados, não há cobrança. Com a cobrança baseada no consumo, você só paga pelo tempo em que seu código é executado, e o Azure escalará conforme necessário. Isso torna o Azure Functions uma opção ideal para APIs.

Principais recursos do Azure Functions

Recursos	Benefícios
Não é necessário gerenciar nenhuma infraestrutura	Permite que você se concentre em agregar valor. O dimensionamento pode ser automatizado e flexível.
Suporte para muitas linguagens populares	<p>Você pode escrever seu código em:</p> <ul style="list-style-type: none"> - C# - JavaScript - F# - Java - PowerShell - Python - TypeScript <p>Para obter detalhes sobre linguagens com suporte no Azure Functions, consulte esta documentação.</p>

Recursos	Benefícios
Experiência de desenvolvimento completa	Ferramentas integradas e recursos de DevOps incorporados, que permitem criar e depurar, bem como implantar e monitorar.
Integração simplificada	Fácil integração com serviços do Azure e ofertas de Software como Serviço (SaaS).
Preços de pagamento por uso	Com o plano de hospedagem por consumo, você só é cobrado quando seu código é executado.

Usando o Azure Functions, você pode criar pequenos itens de funcionalidade rapidamente e hospedá-los em um ambiente elástico que gerencia o dimensionamento de forma automática.

Caso de uso do Azure Functions

- O Azure Functions é uma boa opção quando você tem um código que é acionado por outros serviços do Azure, por eventos baseados na Web ou em uma agenda. Você também pode usar o Azure Functions quando não precisar da sobrecarga de um projeto hospedado completo ou quando desejar pagar apenas pelo tempo em que seu código é executado.
- Para exemplificar ainda mais isso, suponha que você queira automatizar o processo de redimensionamento de imagens sempre que um novo arquivo de imagem for carregado no Armazenamento de Blobs do Azure. É possível criar uma função do Azure que será acionada sempre que um novo arquivo de imagem for carregado no Armazenamento de Blobs do Azure. A função redimensionará a imagem e a gravará na conta de Armazenamento de Blobs. Não há necessidade de gravar a estrutura para se conectar ao Armazenamento de Blobs: basta configurá-la.

Para saber mais

- [Documentação do Azure Functions](#)
- [Conceitos de gatilhos e associações do Azure Functions](#)

Colocar em prática

Assim como o Serviço de Aplicativo do Azure, você pode experimentar o Azure Functions gratuitamente em um ambiente de área restrita, sem uma assinatura do Azure:

1. Navegue até [esta URL](#) e crie sua primeira função do Azure.
2. Marque a caixa de seleção **I'm not a robot** (Não sou um robô) e clique no botão **Create a sample function app** (Criar uma aplicação de função de exemplo):

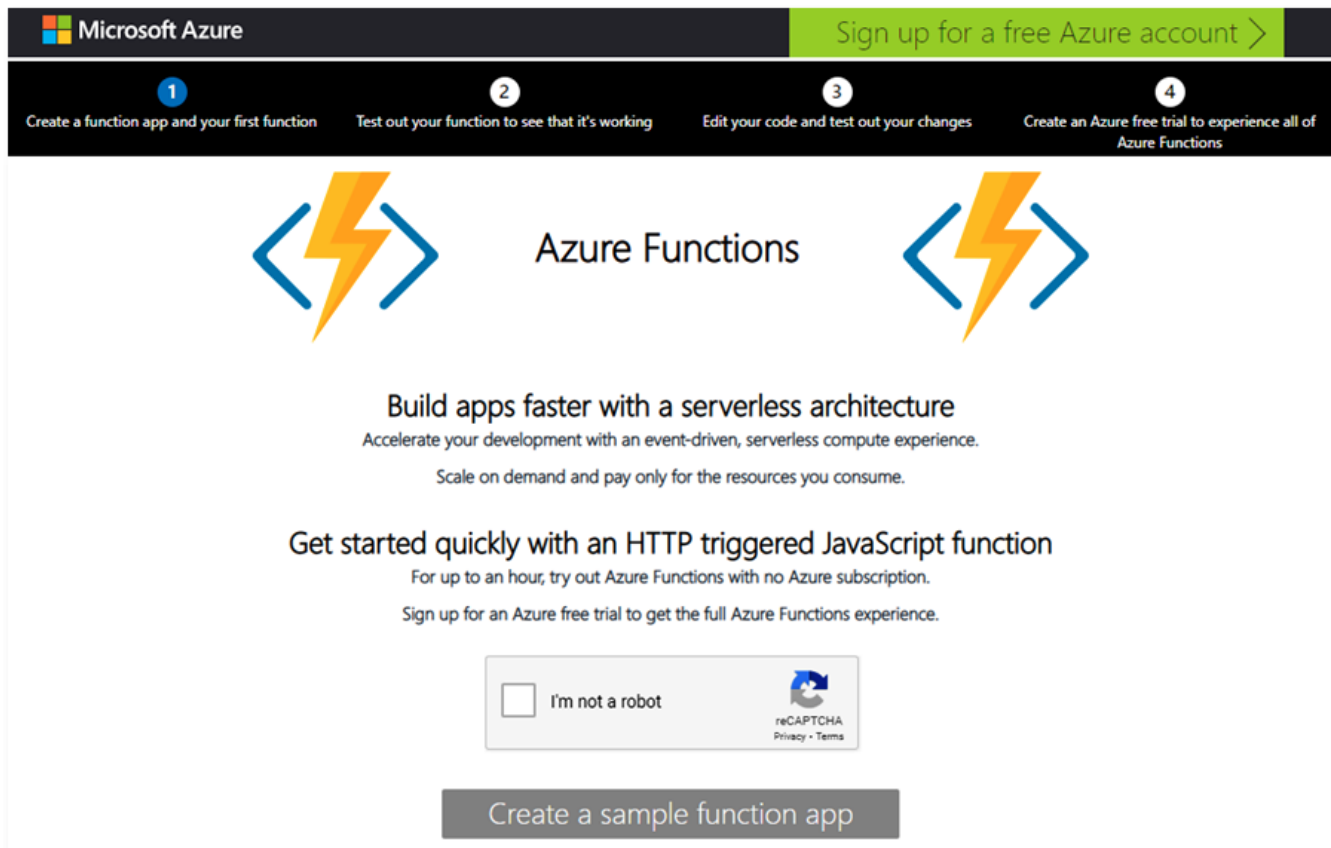


Figura 37: Criar uma aplicação de função de exemplo

- Uma aplicação de função de gatilho HTTP de exemplo é pré-preenchida no painel do editor para você. Essa função é acionada por uma solicitação HTTP GET ou POST e envia como saída uma resposta HTTP com base no código de usuário fornecido. Clique no botão **Run** (Executar) para vê-la em ação:

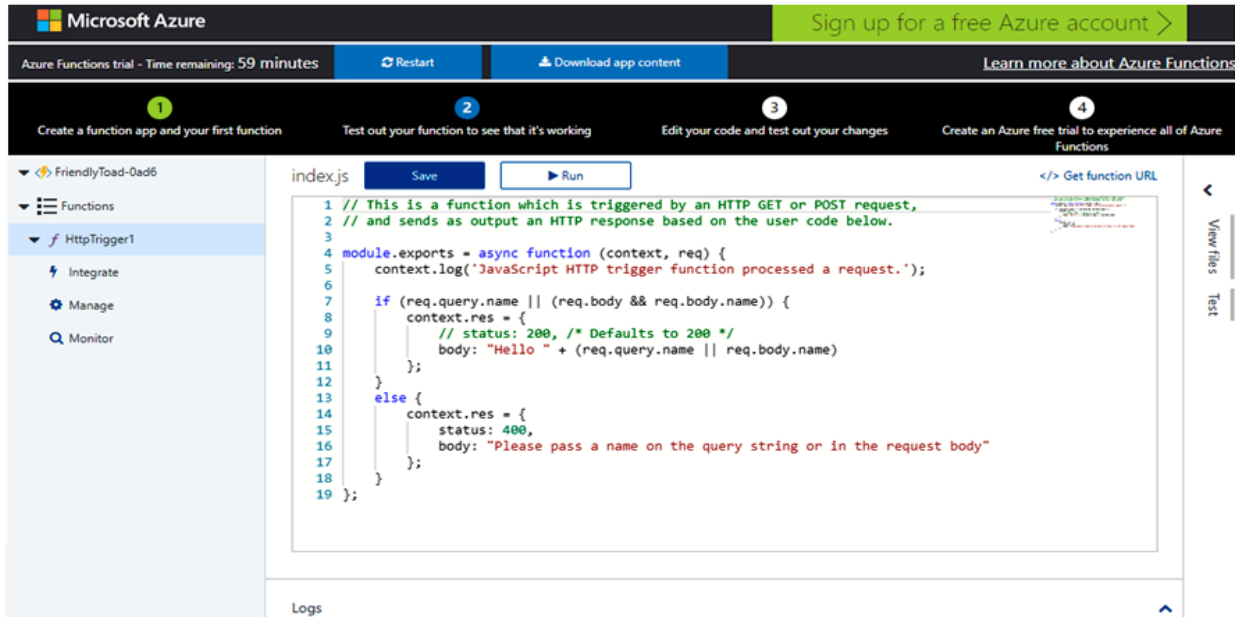


Figura 38: Clicar em Run

- Se a execução for bem-sucedida, você verá a saída a seguir:

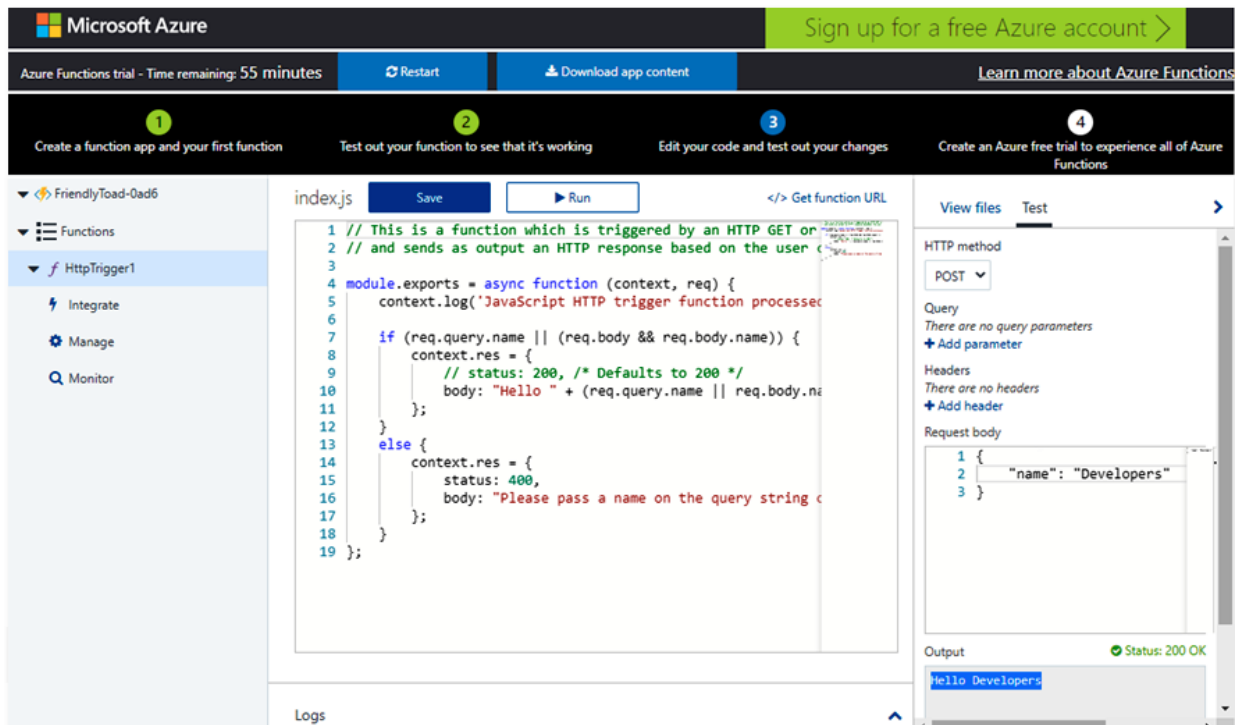


Figura 39: Verificar a saída após a execução

- Agora, altere o **HTTP method** (Método HTTP) para **GET** e forneça o nome e a variável da consulta da seguinte maneira. Observe a saída após clicar no botão **Run**:

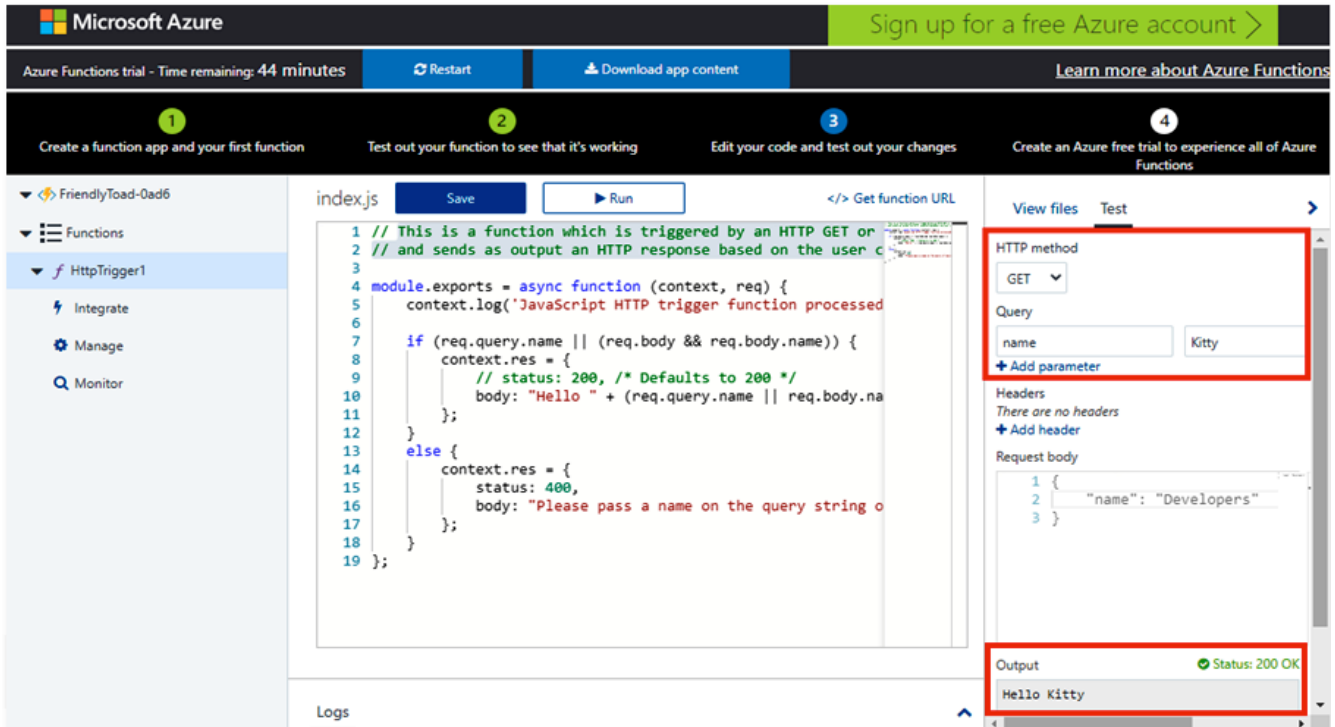


Figura 40: HTTP Get

- Você pode criar outra aplicação simples de prova de conceito do Azure Functions nesta interface gratuita. Experimente!

Saiba mais sobre como [escolher um serviço de computação do Azure para sua aplicação](#). Na próxima seção, forneceremos alguns recursos e materiais de aprendizagem úteis.



Capítulo 5: Mais aprendizagem e recursos

Aprender os conceitos básicos do Azure

Se você é novo no Azure, recomendamos os seguintes roteiros de aprendizagem interativos:

- [Conceitos básicos do Azure](#)
- [Explorar os conceitos de nuvem do Microsoft Azure](#)
- [Distinguir os principais serviços do Microsoft Azure](#)
- [Examinar a segurança, a privacidade, a conformidade e a confiabilidade do Microsoft Azure](#)
- [Analisar os preços, os contratos de nível de serviço e os ciclos de vida do Microsoft Azure](#)
- [Microsoft Learn](#)

Depois de se tornar proficiente no Azure, considere fazer o exame AZ-900 Noções básicas do Microsoft Azure para receber a certificação. Para obter mais informações, visite [a diretriz desse exame](#).

Ferramentas necessárias para desenvolver seu projeto de prova de conceito para o Azure

Veja a seguir uma lista de ferramentas essenciais para os exemplos mostrados neste guia:

- [Assinatura do Azure](#)
- [Visual Studio Code](#)
- [Azure Functions Core Tools](#)
- [Conta do GitHub](#)
- [Navegador Microsoft Edge \(baseado no Chromium\)](#)
- [Node.js](#)

Outros recursos úteis

Introdução às Máquinas Virtuais do Azure

- [Documentação das Máquinas Virtuais Windows](#)
- [Documentação das Máquinas Virtuais Linux](#)

Introdução ao Serviço de Kubernetes do Azure

- [Serviço de Kubernetes do Azure](#)

Introdução ao Serviço de Aplicativo do Azure

- [Ferramenta Migração do Serviço de Aplicativo](#)
- [Visão geral do Serviço de Aplicativo do Azure](#)
- [Try Azure App Service](#)

Introdução ao Azure Functions

- [Documentação do Azure Functions](#)
- [Sobre os conceitos de gatilhos e associações do Azure Functions](#)
- [Experimentar o Azure Functions](#)

Capítulo 2: Projeto de exemplo – implementar uma aplicação Web usando Aplicativos Web Estáticos do Azure

- [Documentação oficial do GitHub Actions](#)
- [Analisar solicitações pull em ambientes de pré-produção em Aplicativos Web Estáticos do Azure](#)

Capítulo 3: Projeto de exemplo – criar um chatbot inteligente

- [SDK do Microsoft Bot Framework](#)
- [Exemplos do Bot Builder](#)
- [QnA Maker](#)

Recursos gerais

- [Escolher um serviço de computação do Azure para sua aplicação](#)
- [Perguntas frequentes sobre a conta gratuita do Azure](#)



Conclusão

Neste guia, discutimos como um projeto de prova de conceito pode ser uma ferramenta valiosa para avaliar se uma tecnologia ou conceito potencial pode ser usado para atender aos requisitos de uma solução de negócios. Ele pode ajudar a identificar possíveis problemas técnicos e logísticos antes da implementação em um projeto base. Além disso, ele fornece insights oportunos sobre a tecnologia e, ao mesmo tempo, reduz os riscos, permitindo que as principais decisões sejam tomadas nos estágios iniciais do processo de desenvolvimento.

No **Capítulo 1:** Guia de prova de conceito, você aprendeu as bases centrais necessárias para planejar e executar um projeto de prova de conceito bem-sucedido.

No **Capítulo 2:** Projeto de exemplo – implementar uma aplicação Web usando Aplicativos Web Estáticos do Azure e no **Capítulo 3:** Projeto de exemplo – criar um chatbot inteligente, mostramos dois projetos práticos que podem inspirá-lo a criar seus próprios projetos de prova de conceito.

No **Capítulo 4:** Uma visão geral do Azure para desenvolvedores, apresentamos a plataforma do Azure. Você aprendeu a obter uma conta gratuita do Azure para começar. Você também aprendeu a escolher o modelo de nuvem e os serviços apropriados para começar a desenvolver seu projeto de prova de conceito para o Azure.

Além disso, fornecemos alguns materiais de aprendizagem úteis no **Capítulo 5:** Mais aprendizagem e recursos.


Agora você está pronto para começar seu projeto de prova de conceito. Boa sorte!

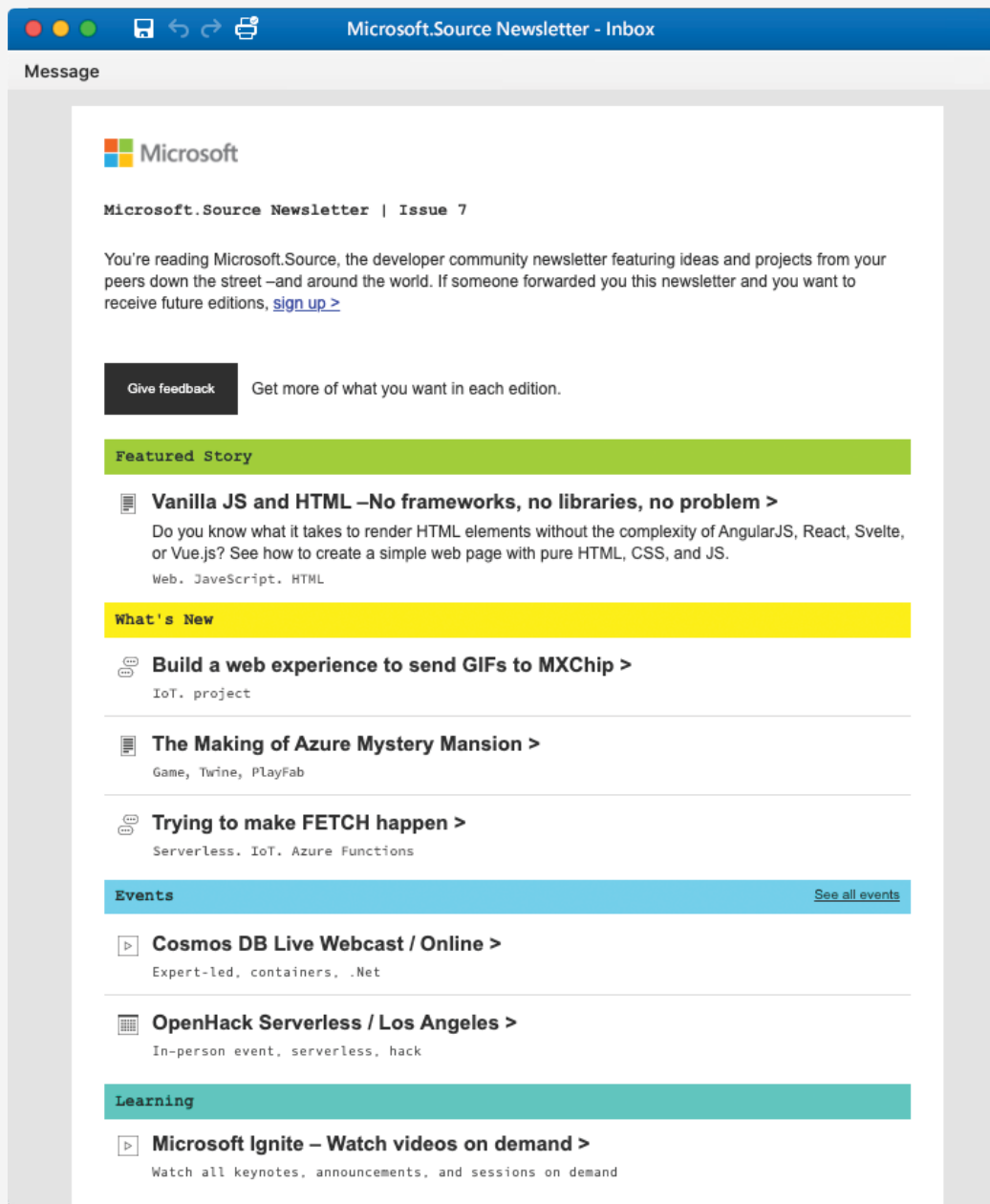


Começar agora

 **Inscreva-se para obter uma conta gratuita do Azure**

 **Saiba mais sobre as soluções do Azure**

 **Fale com um especialista em vendas para receber ajuda em relação a preços, práticas recomendadas e implementação de uma prova de conceito**



Por desenvolvedores,
para desenvolvedores
Boletim informativo
Microsoft.Source

Receba artigos técnicos, código de exemplo e informações sobre os próximos eventos no Microsoft.Source, o boletim informativo mensal da comunidade de desenvolvedores.

- Acompanhe as tecnologias mais recentes
- Conecte-se com seus colegas em eventos da comunidade
- Aprenda com recursos práticos



Inscrição