



Aprenda com quem faz

# Soluções para Desenvolvimento

## Professor Henrique Eduardo Souza

2023



## SUMÁRIO

Capítulo 1. O que é DevOps?.....	6
Como o DevOps funciona na prática? .....	6
O que é o Azure DevOps.....	8
Conhecendo os recursos.....	9
Capítulo 2. Controle de Versão.....	13
Benefícios de se usar Controle de Versão .....	13
Introdução ao Azure Repos.....	15
Capítulo 3. Azure Pipelines .....	22
O que é CI e CD.....	23
Padrões de CD .....	24
O que é DevTest Labs.....	26
Capítulo 4. Azure App Service .....	30
O que é Azure App Service.....	30
Quais Plataformas podemos trabalhar.....	30
O que é Azure APIM? .....	32
Quando usar o Azure APIM?.....	32
Cenários de Utilização: Exposição de APIs.....	32
Cenários de Utilização: Modernização e Migração .....	33
Capítulo 5. Azure Functions.....	35
O que é Serverless e como funciona.....	35
Funcionamento e Escalonamento.....	35
Compreendendo os modelos de bind das Azure Functions .....	36
O que são logic apps.....	37
Capítulo 6. Internet das Coisas (IoT) .....	41
O que é Internet das Coisas (IoT).....	41

Funcionamento do IoT.....	43
Benefícios do IoT.....	49
Segurança e Privacidade.....	50
Conceitos básicos sobre IoT Hub.....	54
AWS IoT Core.....	55
Introdução ao IoT Central.....	59
<b>Capítulo 7. Azure Power Platform.....</b>	<b>61</b>
Power Apps.....	61
Power Automate.....	62
Power BI.....	62
Power Virtual Agents.....	63
<b>Capítulo 8. Ferramentas para Gerenciamento e Configuração de Ambientes no Azure.....</b>	<b>66</b>
Azure Portal.....	66
Azure Mobile App.....	66
Azure PowerShell.....	66
Azure CLI.....	66
Cloud Shell.....	67
Templates ARM.....	67
<b>Capítulo 9. Soluções de Infraestrutura como Código (IaC).....</b>	<b>69</b>
O que é a iac (infraestrutura como código)?.....	69
Evitar a configuração manual para impor consistência.....	69
Fornecer ambientes de teste estáveis rapidamente em escala.....	70
Usar arquivos de definição declarativa.....	71
O que é Ansible.....	71
O que é Terraform.....	72

Capítulo 10. Inteligência Artificial (IA) e Machine Learning.....	74
O Algoritmo.....	76
Alimente-me e será seu fim.....	76
Ciência de dados e Deep Learning.....	77
Águas profundas da aprendizagem.....	77
Capítulo 11. Machine Learning com Plataformas em Nuvem.....	80
Conceitos básicos de Inteligência Artificial.....	80
O que é o aprendizado de máquina?.....	81
Amazon SageMaker.....	83
DataRobot.....	83
Google Cloud Machine Learning Engine.....	84
IBM Watson Studio.....	84
Azure OpenAI.....	85



**XP**e

# > Capítulo 1



## Capítulo 1. O que é DevOps?

---

Para começar, vamos entender o básico. DevOps é uma prática de engenharia de software que consiste em integrar o trabalho de desenvolvedores (Dev) e profissionais de operação (Ops) em TI.

O objetivo é evitar falhas na comunicação entre esses dois times e, assim, aumentar a capacidade de em uma empresa entrar novos softwares e atualização de softwares com rapidez e qualidade.

Por depender de um clima organizacional favorável à integração de diferentes equipes, o DevOps também é conhecido como uma “cultura DevOps”.

O DevOps depende não apenas de uma integração das ferramentas de trabalho de desenvolvedores e profissionais de operação, mas também de uma capacidade de trabalhar em grupo e de criar processos padronizados. É dessa forma que uma empresa de tecnologia consegue evitar retrabalhos, atrasos e projetos desalinhados com os objetivos iniciais.

### Como o DevOps funciona na prática?

A cultura DevOps tem um impacto direto no ciclo de vida de um aplicativo ou de qualquer outro sistema digital em desenvolvimento. Então, vamos entender como funciona esse ciclo para ver como o DevOps atua na prática!

No geral, a cultura DevOps permite que os responsáveis pelas fases de planejamento, desenvolvimento, entrega e operação se mantenham em constante diálogo e entendam como o seu trabalho impacta o trabalho dos demais.

No DevOps, a função de um profissional está envolvida, de alguma forma, em cada fase:

- Planejamento: idealizar e definir as funcionalidades e demais características dos sistemas em desenvolvimento;
- Desenvolvimento: fazer a gravação, o teste e a integração do código para o desenvolvimento do sistema. Também é o momento de fazer a compilação do código em “artefatos de compilação”, que podem ser usados em vários ambientes;
- Entrega: implantar os sistemas digitais em ambientes de produção;
- Operação: monitorar e solucionar possíveis problemas que aparecem nos aplicativos já implementados em ambientes de produção.

Em um mercado tão competitivo quanto o de TI, as empresas buscam profissionais que cumpram o papel de estar à frente das mudanças que precisam ser feitas para a interação entre os setores de Tecnologia da Informação.

A solução que vamos abordar é uma das mais completas para DevOps, ela abrange todos os pilares necessários para adoção da prática do DevOps.

Figura 1 – Azure DevOps.



Fonte: <https://azure.microsoft.com/pt-br/services/devops>.

## O que é o Azure DevOps

O Azure DevOps é uma solução Microsoft para organização de projetos, gerenciamento de atividades, gestão de capacidade, organização de Pipelines CI e CD (iremos abordar ambos durante o módulo) também podemos contar com gestão de cenários de testes, gestão de artefatos e um poderoso repositório Git e/ou TFS.

## Tipos de Assinatura Azure DevOps

Essa é uma dúvida muito comum entre muitos clientes nos quais tenho atuado, sempre ouço as seguintes frases:

- “Azure DevOps é muito caro, prefiro usar o Trello”.
- “Temos 300 usuários que geram histórias para minha fábrica de Software, o custo é inviável”.

É muito comum clientes confundirem o licenciamento do Azure DevOps com licenciamento de Microsoft Visual Studio, realmente o licenciamento da Microsoft para Visual Studio Profissional é algo em torno de 5.300,00 uma licença profissional que nos dá direito a muitos benefícios, inclusive Azure DevOps, porém, para o caso do meu cliente que contratou uma fábrica de softwares para seus 300 Key Users, melhor dizendo, usuários chave para o negócio, esses não gastaria 1 centavo para trabalhar colocando suas histórias e acompanhando seus projetos. Vocês podem estar perguntando agora: como assim Henrique? Você acabou de dizer que uma licença custa R\$5.300,00. Mais à frente estarei demonstrando os modelos e preços de contratação que podem facilmente ajudar na decisão de sua companhia.

Plano Básico	Plano Básico + Test Plans
Azure Pipelines: inclui a oferta gratuita de SERVIÇOS INDIVIDUAIS.	Inclui todas as funcionalidades do plano Básico.



<p>Azure Boards: acompanhamento de item de trabalho e quadros Kanban.</p> <p>Azure Repos: Repositórios Git privados ilimitados.</p> <p>Azure Artifacts: 2 GiB gratuitos por organização.</p>	<p>Testar o planejamento, o acompanhamento e a execução.</p> <p>Testes baseados no navegador com anotação.</p> <p>Execução de teste do cliente avançado.</p> <p>Teste de aceitação do usuário.</p> <p>Relatório centralizado.</p>
<p>Primeiros 5 usuários Gratuitos</p> <p>Após * 29,40 por usuário</p>	<p>Média de * 255,00 mês</p>

Porém, essas informações ainda não são suficientes, lembra-se do cliente que comentou que tinham 300 usuários, e a fábrica de Software que realizaria as atividades? Pois bem, nesses casos temos uma abordagem mais barata ainda, uma vez que contratei uma fábrica de software, o custo de licenças para uso de Azure Repos é, normalmente, por conta do contrato da fábrica de software, logo seus 300 usuários podem utilizar a ferramenta de Boards, organizar toda capacidade de o projeto escrever suas histórias com custo muito baixo.

## Conhecendo os recursos

### Azure Boards

É um dos serviços mais utilizados na ferramenta Azure DevOps, uma vez que o serviço de Azure Boards pode substituir por completo ferramentas como Jira e Trello, o serviço Azure Boards ajuda as equipes a gerenciar seus projetos de software. Ele fornece um conjunto abrangente de recursos, incluindo suporte nativo para Scrum e Kanban, painéis personalizáveis e relatórios integrados.

Você pode iniciar rápida e facilmente o acompanhamento de histórias de usuários, itens de pendências, tarefas, recursos e bugs associados ao seu

projeto. É possível controlar o trabalho adicionando itens de trabalho com base no processo e nos tipos de item de trabalho disponíveis para seu projeto.

### Test Plans

O Test Plans é o único serviço do Azure comercializado com uma licença específica. Esse recurso tem como principal objetivo organizar e executar diversos planos de testes, é possível colocá-lo integrado com o módulo de Pipelines, Boards e Repos, onde podemos integrar nossos testes de unidade codificados pelos desenvolvedores, enquanto desenvolvem as histórias colocadas no Azure Boards, validando os cenários de testes criados no Test Plans. Com a união de todos os recursos podemos executar, acompanhar, rastrear e criar métricas de qualidade do software baseados em todo ciclo de vida do desenvolvimento do software.

### Artifacts

O Azure Artifacts é uma maneira simples e segura de manter as dependências do seu projeto de software de forma simples e pragmática. Quem nunca teve problemas ao iniciar em um projeto em um novo cliente, e havia dezenas de pacotes espalhados na máquina de algum desenvolvedor ou em algum lugar na internet? O Azure Artifacts ajuda a resolver esse problema, é possível controlar o ciclo de vida de cada pacote, validando vulnerabilidades antes mesmo de utilizá-los, o Azure Artifacts gerencia pacotes do tipo Nuget, Maven, PyPi, NPM, Docker.

### Pipelines

O Azure Pipelines é um serviço bem completo, onde usamos para criação no fluxo de CI (integração contínua) que consiste na geração de Build da aplicação em basicamente todas as linguagens atualmente utilizadas no mercado global, e pelo fluxo de CD (implantação contínua), onde a responsabilidade do fluxo é colocar sua aplicação no ar em diversos



ambientes, podendo haver aprovações em determinadas partes do fluxo. Teremos um módulo a frente especificamente para abordar o tema.



**XP**e

## > Capítulo 2



## Capítulo 2. Controle de Versão

---

O controle da versão de código-fonte é uma prática diária essencial. O controle de versão é uma parte comum da rotina do desenvolvedor e, se aproveitado corretamente, pode economizar enormes custos e recursos para as organizações. Embora o controle de versão seja hoje um aspecto de bom senso da programação, é importante ocasionalmente ver por que fazemos, o que fazemos e como o fazemos, esse controle de versão uma vez mal feito pode afetar todo o fluxo de valor em uma organização.

### Benefícios de se usar Controle de Versão

Primeiramente, antes de falar sobre controle de versão, vamos contar um pouco sobre o que é controle de versão. Um sistema de controle de versão permite que os desenvolvedores colaborem no código e rastreiem as alterações. O controle de código-fonte é uma ferramenta essencial para todos os projetos com um ou mais desenvolvedores.

Os sistemas de gerenciamento de controle de versão fornecem um histórico de execução do desenvolvimento de código e ajudam a resolver conflitos ao mesclar contribuições de múltiplos desenvolvedores. Para a maioria das equipes de software, a código-fonte é um repositório de conhecimento e compreensão inestimáveis sobre o domínio do negócio e problemas antigos resolvido em suas aplicações, no qual desenvolvedores coletaram e refinaram por meio de um esforço cuidadoso. O controle de versão protege o código-fonte de catástrofe quanto a degradação casual do erro humano e das consequências não intencionais.

*“Na minha opinião, só existe código a menos que esteja versionado em um controle de versão. Com esses recursos temos o primeiro passo para iniciar o processo de entrega contínua” Henrique Eduardo Souza no Livro DevOps da Vida Real.*

Abaixo segue uma pequena lista de benefícios em utilizar controle de versão:

- Criar fluxos de trabalho: os fluxos de trabalho de controle de versão evitam o caos de desenvolvedores não usarem a mesma versão do código, podendo usar versões antigas e voltando problemas conhecidos pelo usuário.
- Trabalhe com versões: cada versão possui uma descrição em forma de um comentário. Essas descrições podem ajudá-lo a acompanhar as alterações em seu código por versão.
- Colaboração: o controle de versão sincroniza versões e garante que suas alterações não conflitem com outras mudanças de sua equipe.
- Mantém o histórico de mudanças: o controle de versão mantém um histórico de alterações à medida que sua equipe salva novas versões do seu código.
- Reutilização: por que fazer o mesmo duas vezes? A reutilização de código é uma prática comum e torna a construção cada vez mais simples e mais rápida.
- Rastreabilidade: as auditorias não são apenas para diversão, em muitos setores isso é uma questão legal. Todas as atividades devem ser rastreadas e os gerentes devem ser capazes de produzir relatórios quando necessário. Um cenário muito falado recentemente sobre rastreabilidade foi sobre os Sistemas da Urna Eletrônica, nos quais eu trabalhei em uma versão em 2006 e até hoje tem todo histórico de versões e protocolos de segurança incluídos ao longo desses anos.

- Capacidade de gerenciamento: os líderes de equipe podem definir e aplicar fluxos de trabalho e métricas ao longo do ciclo de vida de uma aplicação
- Colaboração: quando as equipes trabalham juntas, a qualidade tende a melhorar.

### Introdução ao Azure Repos

Azure Repos é um conjunto de ferramentas de controle de versão que você pode usar para gerenciar seu código. Sendo seu projeto de software grande ou pequeno, usar o controle de versão o mais rápido possível é uma boa prática.

O Azure Repos fornece dois tipos de controle de versão:

- *Git*: controle de versão distribuído.
- *Team Foundation Version Control (TFVC)*: controle de versão centralizado.

### Github x Git Comandos básicos

Quando falamos Git é importante frisar que foi criado em 2005 por *Linus Torvalds* com intuito de ser um controle de versão inteligente e distribuído, onde é possível gerenciar localmente suas alterações e sincronizar com servidor a versão desejada. A maior vantagem do uso Git é trabalhar em equipe, porém, muitos desenvolvedores utilizam para projetos individuais devido à quantidade de recursos que ele oferece.

Já o GitHub é uma rede social composta por desenvolvedores. A primeira parte do nome, “Git”, foi dada por causa da utilização do sistema de controle de versão, já o nome “Hub” traz muito do conceito de integração e interconexão de pessoas do mundo todo.

Abaixo vamos colocar uma lista de comandos muito utilizados no Git, Github e no Azure DevOps.

- `git help`: comando de ajuda para trazer exemplos em qualquer comando Git.
- `git config`: a primeira coisa que devemos fazer quando instalamos o Git é definir o seu nome de usuário e endereço de e-mail.
  - `git config --global user.name "Henrique Souza"`
  - `git config --global user.email hsouza.eduardo@email.com`
- `git clone`: comando no qual você literalmente clona um repositório remoto para o seu repositório local.
  - `git clone https://seurepositprio.com/repositoriotest/app.git`
- `git add`: toda e qualquer alteração no seu código-fonte, após ser clonado, devem ser incluídas com *git add*. Uma forma comum de usar `git add` é incluído todos os arquivos alterados conforme exemplo abaixo:
  - *git add.*
- `git commit`: armazena as alterações juntamente com uma mensagem de registro onde o usuário descreve as mudanças realizadas no repositório local.
  - `git commit -m "Minhas Alterações"`
- `git push`: o `git push` é o comando que realiza a transferência dos commits a partir do seu repositório local para um repositório remoto.
  - `git push`



- `git pull`: incorpora as alterações de um repositório remoto ao repositório local.
  - `git pull`

### Como usar Git no universo empresarial

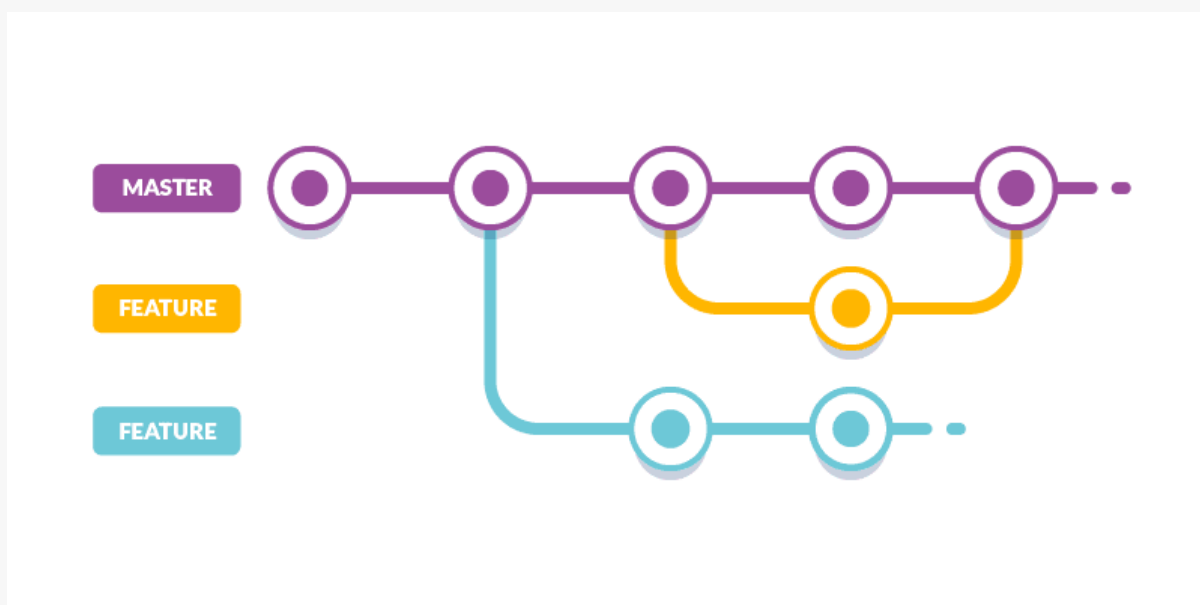
Como um sistema de controle de versão, o Git é fácil de começar, mas difícil de dominar. Enquanto não houver ninguém para implementar o Git da maneira certa, existem várias técnicas que podem ajudá-lo a dimensionar a implementação em toda a organização. Coisas simples como estruturar seu código em micro repositórios, selecionando um modelo enxuto de ramificação e junção através pull requests para revisão de código, podem fazer com que suas equipes se tornem mais produtivas. Existe dois fluxos muito comuns para implementações em ambientes corporativos, o primeiro que iremos abordar será o Feature branch, em seguida abordaremos o GitFlow.

A ideia central por trás do Fluxo de Trabalho por Feature branch é que todo o desenvolvimento de novos recursos ou correções deve ocorrer em um branch, mas antes precisamos explicar o que é branch. Para repositórios de controle de versão o termo branch é uma ramificação do seu código, onde o código original é mantido e é feito uma cópia em uma ramificação chamada de branch. Esse encapsulamento torna mais fácil para vários desenvolvedores trabalharem seus recursos específicos sem alterar a base de código principal. Também significa preservar a branch master com código sempre funcionando com a última versão em produção, o que é uma grande vantagem para ambientes de integração contínua.

O desenvolvimento de feature branches também possibilita aproveitar os pull requests, que são uma maneira de iniciar discussões em torno de uma branch. O pull request é a oportunidade de outros desenvolvedores aprovar um recurso antes de ser integrado ao projeto oficial. Ou seja, se você ficar preso no meio de um recurso, pode abrir uma

solicitação de pull pedindo sugestões de seus colegas. O que quero dizer é que as solicitações pull fazem com que seja incrivelmente fácil para sua equipe comentar sobre o trabalho uns dos outros. Além disso, os Feature Branch podem (e devem) ser enviados para o repositório central. Isso permite compartilhar um recurso com outros desenvolvedores sem tocar em nenhum código oficial. Uma vez que a Branch Master é o único e especial, armazenar várias feature branches no repositório central não apresenta problemas.

Figura 2 – Feature Branch WorkFlow.

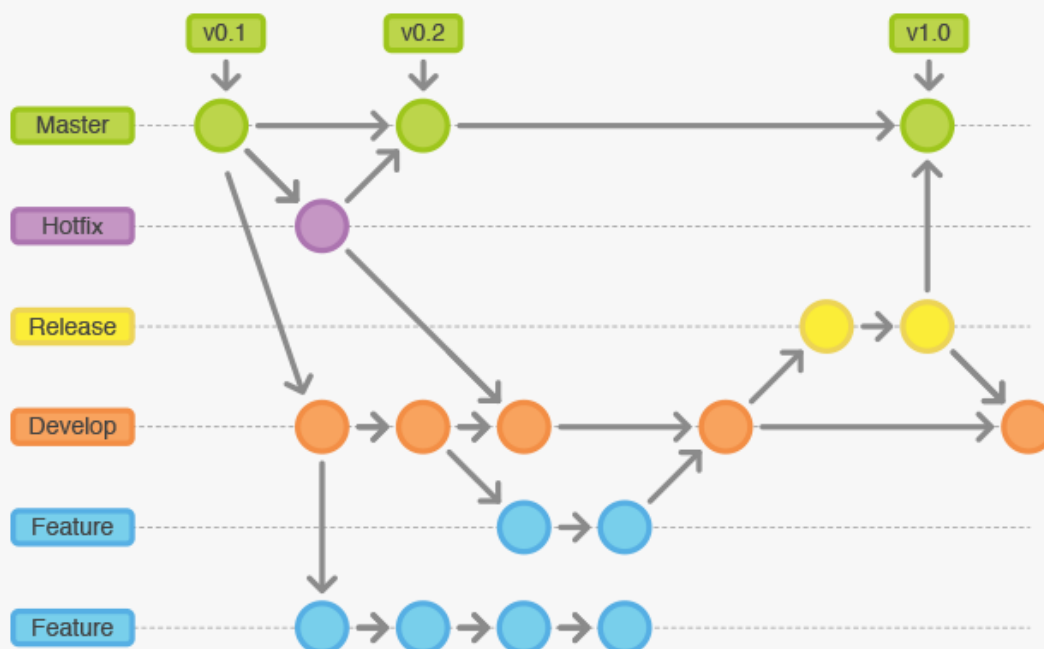


Gitflow é um projeto de fluxo de trabalho Git que foi publicado e popularizado por *Vincent Driessen da nvie*. O fluxo de trabalho Gitflow define um modelo de ramificação estrito projetado em torno do lançamento do projeto. Ele fornece uma estrutura robusta para gerenciar projetos maiores. O Gitflow é ideal para projetos com um ciclo de lançamento agendado. Esse fluxo de trabalho não adiciona nenhum novo conceito ou comando além do que é necessário para o fluxo de trabalho de ramo de recurso. Em vez disso, ele atribui papéis muito específicos para diferentes ramos e define como e quando eles devem interagir. Além de ramos de recursos, ele usa ramos individuais para preparar, manter e gravar

lançamentos. Você também pode aproveitar todos os benefícios do Fluxo de Trabalho do Feature Branch: pull request, experimentos isolados e colaboração mais eficiente.

Além da ideia abstrata de fluxo de trabalho Gitflow, há um conjunto de ferramentas git-flow mais tangível disponível que se integra com Git para fornecer extensões especializadas de ferramenta de linha de comando Gitflow Git.

Figura 3 – GitFlow.



### Trabalhar com Pull Requests

Falamos tanto de pull request e não explicamos como o fluxo funciona. Uma vez que o pedido de pull request é feito ao repositório original, ou uma branch do repositório original, ele fará uma ação de pull (puxar) as atualizações do repositório principal master ou de um branch do próprio repositório, veja a seguir a imagem de um pull request em Azure Repos.

Figura 4 – Exemplo de Pull Request.

The screenshot shows the GitHub 'New Pull Request' interface. At the top, the repository is 'FabrikamFiber / Fabrika...'. The 'Pull Requests' tab is selected. The pull request is from 'FabrikamFiber/jamal.fork' to 'users/jamal/date-fix' into 'FabrikamFiber/master'. The title is 'Added a new option to the settings page'. The description includes a list of changes: 'Added a new option for users to manage delivery preferences. Tested the following: - [ ] Opt out of email - [ ] Opt into email - [ ] Specify an alternate address'. The 'Reviewers' section shows '[FabrikamFiber]\FabrikamFiber Team' as the reviewer. The 'Work Items' section is empty.

**New Pull Request**

FabrikamFiber/jamal.fork → users/jamal/date-fix into FabrikamFiber/master

**Title \***

Added a new option to the settings page

**Add label**

**Description**

Added a new option to the settings page.

Added a new option for users to manage delivery preferences. Tested the following:

- [ ] Opt out of email
- [ ] Opt into email
- [ ] Specify an alternate address

*Markdown supported.*

**Reviewers**

[FabrikamFiber]\FabrikamFiber Team X Search users and groups to add as reviewers

**Work Items**



**XP**e

## > Capítulo 3



## Capítulo 3. Azure Pipelines

---

Atualmente, os negócios exigem entrega contínua de valor, e esse valor é criado apenas quando um produto é entregue a um cliente satisfeito. Não é criado apenas quando um conjunto de grandes entregas no processo é concluído. Isso exige que você redefina o foco das entregas para um fluxo de valor de ponta a ponta.

A ideia central é criar um processo que seja possível repetir, de forma confiável e com melhoria incremental para a obtenção de software, do conceito ao cliente. O objetivo é permitir um fluxo constante de mudanças na produção por meio de uma linha de produção de software automatizada. Pense nisso como um pipeline.

O pipeline divide o processo de entrega de software em estágios. Cada etapa visa verificar a qualidade dos novos recursos de um ângulo diferente para validar a nova funcionalidade e evitar erros que podem afetar seus usuários. O pipeline deve fornecer feedback para a equipe e visibilidade do fluxo de mudanças para todos os envolvidos na entrega do(s) novo(s) recurso(s).

Um pipeline de entrega permite fluxo de mudanças menores com mais frequência, com foco no fluxo. Seus times podem se concentrar na otimização da entrega de mudanças que trazem valor quantificável para o negócio. Essa abordagem leva as equipes a monitorar continuamente e aprender onde estão encontrando obstáculos, resolver esses problemas e melhorar gradualmente o fluxo de entrega. Conforme o processo continua, o ciclo de feedback fornece novos insights sobre novos problemas e obstáculos a serem resolvidos. O pipeline foca no ciclo de melhoria contínua.

## O que é CI e CD

Integração contínua (CI) é o processo de automatizar a construção e teste do código sempre que um membro da equipe confirma as alterações no controle de versão. CI incentiva os desenvolvedores a compartilhar seus códigos e testes de unidade mesclando suas alterações em um repositório de controle de versão compartilhado após a conclusão de cada pequena tarefa.

A confirmação do código aciona um sistema de compilação automatizado para obter o código mais recente do repositório compartilhado e para construir, testar e validar o branch master completo. A ideia é minimizar o custo da integração, tornando-a uma consideração antecipada. Os desenvolvedores podem descobrir conflitos entre o código novo e o existente no início, enquanto os conflitos ainda são relativamente fáceis de reconciliar. Assim que o conflito for resolvido, o trabalho pode continuar com a confiança de que o novo código cumpre os requisitos da base de código existente.

Quando o foco é falar sobre CD, preciso começar com uma pequena história. Há alguns anos, a TI era um departamento facilitador, ela estava lá para oferecer suporte aos usuários de negócios porque o tempo provou que o software desenvolvido tinha má qualidade por padrão, as alterações de software eram um risco. A solução para esse “problema de qualidade” era manter as alterações sob controle escrito. O departamento que passou a ser responsável pelo controle das mudanças passou a ser o departamento de TI Pro. No passado, mas também hoje, o departamento de TI Pro é responsável pela estabilidade dos sistemas, enquanto o departamento de desenvolvimento é responsável pela criação de novo valor.

Essa divisão coloca muitas empresas em uma situação difícil. Departamentos de desenvolvimento são motivados a entregar valor o mais rápido possível para manter seus clientes satisfeitos. Por outro lado, a TI

está motivada para não mudar nada, porque a mudança é um risco, e eles são responsáveis por mitigar os riscos e manter tudo estável. E o que ganhamos com isso? Longos ciclos de liberação. Longos ciclos de lançamento, muitos testes, congelamentos de código, trabalho noturno e durante o fim de semana, e muitas pessoas envolvidas, certifique-se de que tudo funciona. Mas quanto mais mudamos, mais risco isso acarreta, e estamos de volta ao começo, em muitas ocasiões resultando em mais um documento ou processo que deve ser seguido. Isso é o que chamo de desenvolvimento baseado em silos.

Mas os tempos mudaram e precisamos lidar com um novo normal. Nossos clientes exigem softwares modernos, e eles querem tudo para ontem. Se não conseguirmos entregar, eles vão para um concorrente. E a competição é feroz. Com a Internet, sempre temos competição global. Com concorrentes em toda a nossa pilha, mas também com concorrentes que oferecem a melhor ferramenta para um aspecto do software que construímos. Precisamos entregar rápido e que produto que fazemos seja bom. Devemos fazer isso com nossa produção de software sendo barata e com alta qualidade. Para conseguir isso, precisamos de algo como entrega contínua (CD).

## Padrões de CD

### Blue-Green Deployments

Blue-Green Deployments envolvem manter duas versões do ambiente: uma "Blue" (azul), que é a versão atualmente em produção, e uma "Green" (verde), que é a nova versão que você deseja lançar.

Exemplo: imagine que você tem um site de comércio eletrônico. Em vez de atualizar o ambiente em produção diretamente, você cria um ambiente idêntico e faz todas as alterações nele. Uma vez que tudo está testado e pronto, você alterna o tráfego para o novo ambiente "verde".



### Canary Releases

Este padrão envolve lançar a nova versão para um subconjunto menor e específico de usuários antes de torná-la disponível para todos.

Exemplo: você atualizou a funcionalidade de busca no seu aplicativo. Em vez de liberar a atualização para todos os usuários, você primeiro a lança para apenas 5% dos usuários e observa como ela funciona.

### Dark Launching

Dark Launching é quando você lança uma nova funcionalidade, mas a mantém "escondida" dos usuários até que esteja completamente pronta e testada.

Exemplo: você adicionou um chat ao vivo no seu site, mas inicialmente o habilita apenas para os administradores e pessoas da equipe de teste para poderem verificar se tudo está funcionando corretamente antes de lançá-lo para todos os usuários.

### A/B Testing

A/B Testing, ou teste A/B, é uma técnica de comparação entre duas versões de uma página, ou aplicativo, contra uma métrica específica como cliques, conclusão de formulários etc.

Exemplo: você tem duas versões diferentes de uma página de destino e quer saber qual delas converte mais usuários. Metade dos visitantes vê a versão A e a outra metade vê a versão B. Depois, você analisa qual versão teve melhor desempenho.

### Feature Toggles

Feature Toggles, ou interruptores de funcionalidades, permitem que você habilite ou desabilite funcionalidades em tempo real, sem a necessidade de um novo deploy.

Exemplo: Você adicionou uma nova funcionalidade de "recomendações de produtos" no seu site. Usando um feature toggle, você pode ligar ou desligar essa funcionalidade sem precisar fazer um novo deploy.

### O que é DevTest Labs

Azure DevTest Labs permite que os desenvolvedores em equipes gerenciem automaticamente VMs (máquinas virtuais) e recursos PaaS sem aguardar aprovações.

O DevTest Labs cria laboratórios compostos por bases pré-configuradas ou modelos do Azure Resource Manager. Eles têm todas as ferramentas e os programas de software necessários que você pode usar para criar ambientes. Você pode criar ambientes em alguns minutos, em vez de horas ou dias.

### Principais vantagens

Usando o DevTest Labs, você pode testar as versões mais recentes de seus aplicativos executando as seguintes tarefas:

- Provisione rapidamente os ambientes do Windows e do Linux usando modelos e artefatos reutilizáveis.
- Integre facilmente seu pipeline de implantação dos Laboratórios de Teste/Desenvolvimento para provisionar ambientes sob demanda.
- Escale verticalmente seu teste de carga provisionando vários agentes de teste e crie ambientes previamente provisionados para treinamento e demonstrações.

## Principais Características

- **Rápido Provisionamento:** DevTest Labs acelera o processo de configuração do ambiente de desenvolvimento e teste, permitindo o uso de modelos pré-configurados.
- **Controle de Custo:** oferece recursos para controlar e otimizar os custos, como auto desligamento de VMs (Máquinas Virtuais) e limites de gastos.
- **Integração com Ferramentas DevOps:** é fácil integrar DevTest Labs com ferramentas de CI/CD (Integração Contínua/Entrega Contínua) para um fluxo de trabalho mais ágil.
- **Isolamento e Segurança:** permite criar ambientes isolados para testar novas funcionalidades e patches de segurança sem afetar o ambiente de produção.
- **Personalização:** os usuários podem criar suas próprias imagens de máquinas virtuais ou utilizar imagens pré-configuradas disponíveis no catálogo do serviço.

## Exemplo Prático

Imagine que você trabalha em uma empresa que desenvolve um aplicativo de comércio eletrônico. Seu time precisa constantemente de ambientes para desenvolver novas funcionalidades, realizar testes e até simular o ambiente de produção.

1. Você pode criar um laboratório no DevTest Labs e adicionar VMs com as configurações que imitam o ambiente de produção.
2. Os desenvolvedores podem rapidamente criar clones dessas VMs para desenvolver ou testar novas funcionalidades.

3. Ao terminar o dia de trabalho, as VMs podem ser configuradas para desligar automaticamente, economizando custos.
4. Antes de mover as alterações para o ambiente de produção, você pode realizar todos os testes necessários em um ambiente que é quase idêntico ao de produção, reduzindo os riscos associados ao lançamento.



**XP**e

## > Capítulo 4



## Capítulo 4. Azure App Service

---

### O que é Azure App Service

O Serviço de Aplicativo do Azure (*Azure App Service*) é um serviço baseado em HTTP para hospedar aplicativos da web, APIs REST e back-end.

O Serviço de Aplicativo não apenas adiciona o poder do Microsoft Azure ao seu aplicativo, como também o de segurança, carga balanceamento, escalonamento automático e gerenciamento automatizado. Você também pode tirar proveito de seus recursos de DevOps, como implantação contínua do Azure DevOps, GitHub, Docker Hub e outras fontes, gerenciamento de pacotes, ambientes de teste, domínio personalizado e certificados TLS / SSL.

Com o Serviço de Aplicativo, você paga pelos recursos de computação do Azure que usa. Os recursos de computação que você usa são determinados pelo plano de serviço de aplicativo em que você executa seus aplicativos.

### Quais Plataformas podemos trabalhar

Você pode desenvolver em sua linguagem favorita, seja .NET, .NET Core, Java, Ruby, Node.js, PHP ou Python. Os aplicativos são executados e escalonados com facilidade em ambientes baseados em Windows e Linux.

### Camadas de preço

Existem 5 (cinco) níveis de camadas de preço:

- Camada de testes: nesta camada de preço os recursos são limitados, podendo hospedar apenas 10 aplicações, tendo 1GB de espaço em

disco sem nenhum custo, porém o tempo de processamento diário é de no máximo 60 min dia.

- Camada Básica: nesta camada de preço os recursos são cobrados o valor de R\$0,089/hora, tendo número ilimitado de aplicações, 10GB de espaço em disco, podendo realizar estabilidade manual de até 3 instâncias, podendo trabalhar com domínios personalizados.
- Camada Padrão: nesta camada de preço os recursos são cobrados o valor de R\$0,466/hora, tendo número ilimitado de aplicações, 50GB de espaço em disco, podendo realizar estabilidade automática e manual de até 10 instâncias, podendo trabalhar com domínios personalizados e habilitando opção de acesso a recursos em diferentes redes.
- Camada Premium: nesta camada de preço os recursos são cobrados o valor de R\$0,544/hora, tendo número ilimitado de aplicações, 250GB de espaço em disco, podendo realizar estabilidade automática e manual de até 30 instâncias, podendo trabalhar com domínios personalizados, habilitando opção de acesso a recursos em diferentes redes e possibilitando integração com on-premisses.
- Camada Isolada: nesta camada de preço os recursos são cobrados o valor de R\$0,466/hora, tendo número ilimitado de aplicações, 1TB de espaço em disco, podendo realizar estabilidade automática e manual de até 100 instâncias, podendo trabalhar com domínios personalizados, habilitando opção de acesso a recursos em diferentes redes e possibilitando integração com on-premisses e o tipo de computação é isolada.

### O que é Azure APIM?

O Azure API Management é um serviço da Microsoft Azure que oferece recursos para criar uma camada de front-end para suas APIs. Este serviço permite que você defina uma API única e consistente, mesmo se os serviços back-end estiverem fragmentados e espalhados por diferentes locais. Além de unificar APIs, ele oferece funcionalidades como autenticação, autorização, limitação de taxa, caching e coleta de métricas/analytics. Em resumo, APIM atua como um gateway de API que gerencia e protege os serviços back-end.

### Quando usar o Azure APIM?

O Azure APIM deve ser usado quando há a necessidade de um controle mais refinado sobre o acesso às suas APIs. Ele é especialmente útil quando você tem várias APIs e deseja consolidá-las em um único ponto de acesso. Se você precisa impor políticas de uso, realizar transformações no payload dos dados, ou precisa de análise detalhada do uso da API, então o Azure APIM é uma excelente escolha. Ele também é ideal quando você está trabalhando em um ambiente empresarial onde a governança, auditoria e segurança são cruciais.

### Cenários de Utilização: Exposição de APIs

Se você tem serviços back-end, como bancos de dados, workflow e outros microsserviços, o Azure APIM permite que você exponha esses serviços como APIs RESTful ou SOAP de forma segura e escalável. Isso é particularmente útil para empresas que desejam oferecer suas funcionalidades como um serviço, seja para uso interno ou como uma oferta para terceiros. O APIM permite que você transforme seus serviços em



produtos, controle quem tem acesso a eles, e até mesmo como esse acesso é concedido.

### Cenários de Utilização: Modernização e Migração

Se você tem uma arquitetura legada e deseja modernizá-la, o Azure APIM pode ajudar na transição, permitindo que você crie uma camada moderna de API em frente aos sistemas legados. Isso facilita a migração gradual, já que novas aplicações podem interagir com o sistema legado através da nova camada de API, enquanto as aplicações antigas ainda podem acessar o sistema legado diretamente. Isso também é útil para empresas que estão fazendo uma transição para a nuvem, permitindo que partes do sistema se movam para a nuvem enquanto outras permanecem no local.



**XP**e

# > Capítulo 5



## Capítulo 5. Azure Functions

---

### O que é Serverless e como funciona

Serverless não significa a ausência de servidores; em vez disso, é um paradigma que tira a responsabilidade do desenvolvedor de gerenciar servidores. Em um ambiente serverless, a infraestrutura subjacente, incluindo a alocação e provisionamento de servidores, é totalmente gerenciada pelo provedor de serviços em nuvem. Isso permite que os desenvolvedores se concentrem exclusivamente no código e nas funcionalidades, sem se preocupar com a infraestrutura.

### Funcionamento e Escalonamento

O serverless opera com base em eventos, o que significa que o código é executado em resposta a gatilhos ou eventos específicos. Isso pode incluir eventos HTTP, alterações em um banco de dados, ou até mesmo um novo arquivo adicionado a um sistema de armazenamento. Além disso, o escalonamento é feito automaticamente. Se uma função serverless tiver alta demanda, o provedor de nuvem aloca automaticamente mais recursos para atender a essa demanda, e vice-versa.

### Vantagens de Custo e Eficiência

Uma das maiores vantagens do serverless é o modelo de precificação. Em vez de pagar por servidores ociosos, você paga apenas pelo tempo de execução do seu código. Isso torna o serverless uma opção muito eficiente em termos de custo para aplicações com padrões de tráfego variáveis. Além disso, o tempo de colocação no mercado (*time-to-market*) é geralmente mais rápido, uma vez que os desenvolvedores podem focar no desenvolvimento de funcionalidades em vez de gerenciamento de infraestrutura.

## Desafios e Limitações

Apesar de suas muitas vantagens, o serverless não é a solução para todos os problemas. Há limitações quanto ao tempo de execução, uso de memória e até mesmo o tipo de tarefas que podem ser executadas eficazmente em uma arquitetura serverless. Além disso, o depuramento e o monitoramento podem ser mais complicados em um ambiente serverless, dado que você tem menos controle sobre o ambiente de execução.

## Casos de Uso Comuns

Serverless é especialmente útil para cenários onde a demanda é imprevisível ou altamente variável. Isso inclui APIs REST, microsserviços, automações, processamento de dados em tempo real e até mesmo tarefas agendadas. Também é popular em arquiteturas orientadas a eventos, onde pequenas peças de código são executadas em resposta a eventos, como cliques, atualizações de banco de dados ou novos uploads de arquivo.

## Compreendendo os modelos de bind das Azure Functions

Vamos abordar uma Introdução ao Azure Functions, onde permite desenvolver aplicativos sem servidor no Microsoft Azure. Você pode escrever apenas o código que precisa para resolver o problema em questão, sem se preocupar com um aplicativo inteiro ou a infraestrutura para executá-lo.

## O que posso fazer com funções?

Azure Functions é uma ótima solução para processamento de dados, integração de sistemas, integração com internet das coisas (IoT) e na construção de APIs e Microsserviços simples. Considere funções para tarefas como processamento de imagem ou processamento de pedidos, manutenção de arquivos ou para qualquer tarefa que você deseja executar em uma programação simples. Azure Functions fornecem modelos para você começar com os diversos cenários.

O Azure Functions oferece suporte a gatilhos, gatilhos estes que são a maneira de iniciar a execução de seu código, e associações, que são maneiras de simplificar a codificação de dados de entrada e saída.

Azure Functions se integram a vários serviços do Azure e serviços de terceiros. Esses serviços podem acionar sua Azure Function e iniciar a execução, ou eles podem servir como entrada e saída para seu código. O seguinte serviços de integrações são suportadas pelo Azure Functions:

- Azure Cosmos DB;
- Azure Event Hubs;
- Azure Event Grid;
- Azure Notification Hubs;
- Azure Service Bus (filas e tópicos);
- Armazenamento do Azure (blob, filas e tabelas);
- On-Premises (usando Service Bus).

### O que são logic apps

Aplicativos Lógicos, ou Logic Apps conforme conhecidos no Azure, é um serviço de nuvem que ajuda a agendar, automatizar e orquestrar tarefas de processos de negócios e na criação de workflows quando você precisa integrar aplicativos, dados, sistemas e serviços nas empresas ou organizações. Logic Apps simplifica como você projeta e constrói soluções escaláveis para integração de aplicativos, de dados, de sistemas, de aplicativos empresariais business-to-business (B2B), seja na nuvem, no local ou em ambos.

Por exemplo, aqui estão apenas alguns cenários em que você pode automatizar com Logic Apps:

- Processe e roteie pedidos em sistemas locais e serviços em nuvem.
- Mova os arquivos carregados de um servidor SFTP ou FTP para o Armazenamento do Azure.
- Envie notificações por e-mail com o Office 365 quando eventos acontecerem em vários sistemas, aplicativos e serviços.
- Crie Webhook de notificações quando necessário estimular um sistema com alguma informação relevante.

Para construir soluções de integração empresarial com Logic Apps, você pode escolher em uma galeria crescente com centenas de conectores prontos para usar, que incluem serviços como Azure Service Bus, Azure Functions, Azure Storage, SQL Server, Office 365, Dynamics, Salesforce, BizTalk, SAP, Oracle DB, ServiceNow, SAP, entre outros. Os conectores fornecem gatilhos, ações ou ambos para a criação de Logic Apps que acessam e processam dados em tempo real.

### Como funcionam os aplicativos lógicos?

Todo fluxo de trabalho de aplicativo lógico começa com um gatilho, que dispara quando um evento específico acontece ou quando é novo, os dados disponíveis atendem a critérios específicos. Muitos gatilhos fornecidos pelos conectores em Logic Apps incluem recursos básicos de agendamento para que você possa configurar a regularidade com que suas cargas de trabalho são executadas. Para agendamento mais complexos ou recorrências avançadas, você pode usar um gatilho de recorrência como a primeira etapa em qualquer fluxo de trabalho.

Cada vez que o gatilho é disparado, o mecanismo de Logic Apps cria uma instância de Logic Apps que executa as ações no workflow. Essas ações



também podem incluir conversões de dados e controles de fluxo, como instruções, switch, loops e if. Por exemplo, este aplicativo lógico começa com um Dynamics 365 disparando com os critérios internos “Quando um registro é atualizado”. Se o gatilho detecta um evento que corresponde a esses critérios, o gatilho dispara e executa as ações do workflow.



**XP**e

## > Capítulo 6





## Capítulo 6. Internet das Coisas (IoT)

---

### O que é Internet das Coisas (IoT)

Internet das Coisas trata-se dos bilhões de dispositivos espalhados por todo o mundo que estão conectados à internet e que conversam entre si. Isso permite reunir informações, analisá-las e, a partir disso, criar ações oriundas do processamento de dados, agora, em volume muito maior do que quando a internet circula apenas pelos computadores, celulares e tablets.

Em resumo, a IoT possibilita a conexão integral 24 horas por dia, desde o celular até o smartwatch, passando pelos aparelhos de assistência virtual, televisores, geladeiras e até mesmo lâmpadas.

A Internet das Coisas pode ser trafegada por uma coleção de serviços de nuvem gerenciados pelos provedores de nuvem (Microsoft, AWS, Google etc.) que conectam, monitoram e controlam até bilhões de ativos de IoT.

Simplificando, uma solução de IoT é composta por um ou mais dispositivos IoT, e por um ou mais serviços de back-end em execução na nuvem, os quais se comunicam entre si. Descrevemos os três componentes abaixo.

- Dispositivos IoT: os dispositivos IoT são compostos por uma placa de circuito eletrônico com sensores conectados ou atuadores que se conectam à Internet. Os dispositivos IoT incluem alguma forma de mecanismo de rádio para se conectar à Internet (como Bluetooth ou Wi-Fi). Sensores de pressão, sensores de temperatura e umidade e acelerômetros são exemplos de dispositivos IoT, eles capturam informações sobre um ambiente específico onde estão localizados.

Por exemplo, um acelerômetro pode indicar se o elevador está em repouso ou em movimento.

- **Comunicação:** seu dispositivo pode se comunicar com serviços de back-end em ambas as direções. O dispositivo pode enviar dados para o serviço. Ele também pode receber uma indicação do serviço para aumentar a frequência da coleta de dados. A comunicação com dispositivos IoT tem algumas considerações adicionais. Por exemplo, os dispositivos IoT podem ter energia limitada e a conectividade pode ser lenta ou intermitente.
- **Serviços de back-end:** os serviços de back-end recebem dados de telemetria em escala dos dispositivos e determinam como processar e armazenar esses dados. Eles também se comunicam com o dispositivo, gerenciando seu ciclo de vida, monitorando e controlando seu estado.

O que os serviços de IoT nos provedores em nuvem permitem a você:

- Reimaginar seus processos de negócios, reduzindo os silos de dados entre clientes, operações, produtos/ativos e funcionários.
- Implantar software como serviço gerenciado para IoT tendo apenas o mínimo de experiência em nuvem (soluções de SaaS).
- Personalizar soluções de IoT específicas do setor para cenários comuns de IoT: soluções de PaaS por meio dos aceleradores de solução de IoT.
- Estender a inteligência da nuvem para seus dispositivos de borda utilizando Edge Computing em nuvem.
- Conectar, monitorar e controlar bilhões de ativos de IoT usando monitoramento em tempo real.

- Criar um modelo digital de seu espaço físico ou de seus ativos usando os Gêmeos Digitais.
- Explorar e receber informações dos dados de IoT de série temporal em tempo real.
- Criar e conectar dispositivos baseados em Node MCU altamente seguros usando os dispositivos homologados.
- Fornecer contexto geoespacial para dados usando as plataformas de Mapas.
- Navegar pela arte do possível além dos aplicativos tradicionais, como manutenção preditiva.
- Gerar um loop de comentários automatizado entre o dispositivo e a nuvem, para aumentar a eficiência da solução.

### Funcionamento do IoT

Antes de explorarmos o funcionamento do IoT em plataformas de nuvem, vamos primeiro recapitular os serviços básicos de computação em nuvem.

A computação em nuvem envolve o fornecimento de serviços hospedados pela Internet. Os serviços de nuvem hospedados são divididos em três categorias: IaaS (Infraestrutura como Serviço), PaaS (Plataforma como Serviço) e SaaS (Software como Serviço). A IaaS fornece recursos de computação virtualizados pela Internet. Normalmente, esses são recursos de hardware implantados em data centers. Exemplos de recursos de IaaS incluem serviços de armazenamento, backup e recuperação. Os serviços de PaaS são focados na simplificação do desenvolvimento de software. A PaaS pode fornecer serviços como gerenciamento de versão e teste. A PaaS

fornece uma estrutura para que os desenvolvedores criem e implantem aplicativos baseados em nuvem rapidamente. No caso do SaaS, o software é hospedado centralmente e gerenciado pelo provedor de nuvem. O usuário final usa o software como serviço. IaaS, PaaS e SaaS se sobrepõem mutuamente. A IaaS e o SaaS precisam, respectivamente, do maior e do menor nível de gerenciamento do usuário final.

O portfólio de produtos do Azure IoT apresenta dois caminhos para a criação de soluções, ou seja, soluções PaaS e soluções SaaS. Você também tem acesso às tecnologias de PaaS subjacentes para desenvolver uma solução granular.

### Soluções de plataforma como serviço

Os aceleradores de solução do Azure IoT são soluções de PaaS personalizáveis para mercados verticais de um setor específico. Os aceleradores de solução de IoT fornecem soluções predefinidas que podem ser implantadas rapidamente em um mercado vertical. Os exemplos de aceleradores de solução de IoT incluem monitoramento remoto, fábrica conectada, manutenção preditiva e simulação de dispositivo.

### Soluções de software como serviço

As soluções de SaaS são adequadas para organizações com menos modelos de dispositivos, com cenários previsíveis e recursos limitados de IoT/TI. O Azure IoT Central é uma solução de SaaS gerenciada, que permite que você se familiarize rapidamente com experiência mínima em IoT. O IoT Central é adequado quando as necessidades de IoT são simples e a experiência em desenvolvimento de nuvem necessária é relativamente pequena.

### Tecnologias de plataforma como serviço

As tecnologias de PaaS fornecem uma opção granular para controlar todos os aspectos da sua solução de IoT, permitindo que você crie uma

solução totalmente personalizada. Você pode classificar as tecnologias de PaaS em quatro categorias: suporte a dispositivos; IoT; dados e análise; visualização e integração. Descrevemos abaixo, resumidamente, cada uma dessas quatro categorias e seus subcomponentes.

### 1) Suporte a dispositivos

O Azure IoT fornece vários meios para dar suporte a uma variedade de dispositivos. Isso inclui o SDK do dispositivo IoT do Azure, os dispositivos IoT certificados do Azure, o Programa de Segurança para o Azure IoT e a IoT do Windows 10.

### 2) IoT

O conjunto de tecnologias de IoT inclui:

- **Hub IoT do Azure:** o Hub IoT do Azure é um serviço gerenciado que permite comunicações bidirecionais entre milhões de dispositivos IoT e um back-end da solução. O Hub IoT permite que você conecte, gerencie e dimensione bilhões de dispositivos IoT da borda para a nuvem.
- **Azure Time Series Insights:** o Azure Time Series Insights permite que você armazene, visualize e consulte grandes quantidades de dados de série temporal gerados por dispositivos IoT para obter insights acionáveis.
- **Azure Sphere:** o Azure Sphere é uma solução de ponta a ponta para proteger dispositivos baseados em MCU (unidade de microcontrolador). O Azure Sphere consiste em três partes: Sistema Operacional do Azure Sphere, MCUs Certificados do Azure Sphere e Serviço de Segurança do Azure Sphere:

- O Sistema Operacional do Azure Sphere é um sistema operacional seguro que cria uma plataforma protegida para dispositivos IoT.
- Os MCUs Certificados do Azure Sphere fornecem conectividade e raiz de confiança de hardware confiável.
- O Serviço de Segurança do Azure Sphere agencia a relação de confiança para comunicação de dispositivo para dispositivo e de dispositivo para nuvem. Ele detecta ameaças emergentes e renova a segurança do dispositivo.

Assim, o Azure Sphere permite que você crie dispositivos e conecte os equipamentos existentes com segurança, usando recursos de segurança de ponta a ponta dispostos em camadas na pilha do dispositivo.

- Azure IoT Edge: o IoT Edge é criado sobre o Hub IoT. Ele permite que você mova partes de sua carga de trabalho para o dispositivo de borda e crie uma experiência melhor para seus clientes.
- Serviço de Provisionamento de Dispositivos no Hub IoT: o Serviço de Provisionamento de Dispositivos no Hub IoT do Azure é um serviço auxiliar para o Hub IoT. O Serviço de Provisionamento de Dispositivos permite que os clientes provisionem automaticamente milhões de dispositivos de maneira segura e escalonável.
- Gêmeos Digitais do Azure: os Gêmeos Digitais do Azure permitem que você modele os relacionamentos e as interações entre pessoas, espaços e dispositivos. Essa capacidade permite criar modelos de um ambiente físico que você está interessado em analisar. Depois que o ambiente tiver sido modelado, você poderá executar algoritmos preditivos para prever o comportamento de todo o sistema. Por exemplo, se um determinado escritório tiver sido

modelado por meio dos Gêmeos Digitais, você poderá otimizar o uso de espaço disponível nesse escritório replicando o espaço físico.

- **Azure Mapas:** o Azure Mapas é uma coleção de serviços geoespaciais que usa novos dados de mapeamento para fornecer um contexto geográfico preciso para aplicativos Web e móveis. Você pode adicionar análise espacial, serviços de mapeamento e serviços de mobilidade ao seu aplicativo usando APIs.

### 3) Dados e análises

**Azure Machine Learning:** o Azure Machine Learning representa um conjunto de produtos para criar, treinar e implantar modelos de machine learning no Azure. As soluções de análise e dados do Azure representam um conjunto de soluções para armazenar e analisar dados de IoT. Isso inclui Azure Stream Analytics, Azure HDInsight, Azure Data Lake, Azure Cosmos DB e Azure Data Fabric. Você também pode criar seu sistema usando o Azure Functions para habilitar um aplicativo sem servidor. O Azure Functions permite que você execute pequenos trechos de código (também chamados de funções) na nuvem. Essas funções podem então ser combinadas para criar um aplicativo mais abrangente. Você paga apenas pelo tempo durante o qual seu código é executado (aplicativo sem servidor).

### 4) Visualização e integração

As soluções de visualização e integração para soluções de IoT incluem Microsoft Flow, Aplicativos Lógicos do Azure, Aplicativo Web do Azure, Hubs de Notificação, Azure Active Directory, Microsoft Power BI, Azure Monitor e Power Apps.

### Como os componentes do Azure IoT se unem para resolver o cenário

Considere o cenário apresentado a seguir sobre a garantia de entrega de alimentos frescos.

O aplicativo de telemetria básico não atende aos requisitos mais complexos. O registro da temperatura de frutas indica apenas os problemas a posteriori. Se houver um problema com uma fruta específica, você precisará saber o local exato e garantir que as outras frutas no palete ainda possam ser consumidas com segurança. Você precisa da capacidade de inspecionar visualmente as frutas remotamente usando uma câmera e indicar frutas que apresentem qualquer problema. Se houver degradação, você precisará saber a distância restante entre o caminhão e seu destino para que você possa informar o varejista e o produtor. Caso o pedido seja urgente, você também precisará notificar veículos próximos para fornecerem ao varejista, se isso for possível.

Por fim, você precisa manter os dados sobre a consignação para fins de seguro e auditoria. Ao analisar os dados, você pode antecipar problemas futuros antes que eles ocorram. Agora, os clientes podem receber dados sobre as frutas, por exemplo, a pegada de carbono.

O Azure IoT ajuda a resolver problemas discutidos nesse cenário, fornecendo um conjunto de produtos que ajudam a implantar a IoT em escala. Todas as mensagens são recebidas e gerenciadas centralmente. Novos dispositivos podem ser provisionados e gerenciados para permitir que diferentes tipos de produtores e varejistas se conectem ao sistema. Os dados de sensores podem ser capturados de várias maneiras (por exemplo, dados de série temporal, de imagem e de vídeo). Se necessário, os dados podem ser gerenciados em dispositivos de borda. Os dados podem ser visualizados na nuvem. A análise pode ser aplicada na nuvem para computar métricas de negócios, como a pegada de carbono. Você também pode usar a análise para calcular anomalias em leituras de sensor e aplicar algoritmos de previsão para prever problemas com base em dados anteriores. Você poderia correlacionar os dados e a análise das informações geográficas.



No entanto, é possível adotar uma abordagem ainda mais inovadora. Você pode repensar os processos de negócios existentes da perspectiva da nuvem inteligente e da borda inteligente. Com base no cenário acima, a implantação da nuvem inteligente e da borda inteligente pode transformar processos existentes de quatro maneiras:

- *Impacto sobre a equipe:* os motoristas de caminhão (que estão geralmente em movimento) agora têm informações mais atualizadas e suporte com base no compartilhamento de dados e informações de mapeamento.
- *Impacto sobre os clientes:* os clientes (varejistas) agora obtêm informações mais atualizadas, com garantia de frutos com uma melhor qualidade.
- *Envolver os clientes:* a pegada de carbono e outras métricas estimulam o envolvimento dos clientes.
- *Otimizar as operações:* todo o processo de entrega de frutos frescos pode ser otimizado por meio de um loop de comentários da implantação de IoT, dividindo os silos de dados.

Esse novo processo poderia gerar um loop de comentários automatizado entre o dispositivo e a nuvem, para aumentar a eficiência da solução.

### Benefícios do IoT

Compreendido o conceito de IoT na indústria (ou IIOT), podemos imaginar que ela traz muitas vantagens, certo? O que a IoT traz de benefício para a indústria? Listamos a seguir:

- Menos custos: um dos benefícios da IoT na indústria é a redução de desperdícios, atrasos e falhas. Essa melhoria nos processos fabris provoca uma redução de custos, além de maior lucro. A IIoT também informa sobre a necessidade de manutenção nas máquinas, diminuindo o gasto com aquisição de novos equipamentos.
- Tomada de decisões fundamentada: com a interconexão de todo o processo fabril, os executivos conseguem ter muitos dados relevantes. A partir da análise das informações, têm uma visão global do desempenho da indústria e podem tomar melhores decisões.
- Menos acidentes: o controle das máquinas à distância, possibilitado pela IoT na indústria, aumenta a segurança dos colaboradores. Com a menor exposição ao risco, há menos acidentes. É um dos grandes benefícios da IoT na indústria.
- Diagnósticos aprimorados: os sensores nas máquinas possibilitam detecção e análise de fatores internos e externos da produção. Isso permite uma análise mais completa do processo.
- Controle da produção: a IIoT promove a comunicação das máquinas ao servidor central em tempo real. Isso traz maior controle da produção a partir dos dados fornecidos.

### Segurança e Privacidade

A Internet das Coisas (IoT) apresenta desafios únicos de segurança, privacidade e conformidade para as empresas no mundo todo. Ao contrário da tecnologia cibernética tradicional, onde essas questões giram em torno do software e como ele é implementado, a IoT diz respeito ao que acontece quando os mundos cibernético e físico convergem. Proteger as soluções de IoT exige a garantia de provisionamento seguro de dispositivos,

conectividade segura entre esses dispositivos e a nuvem, e proteção de dados segura na nuvem durante o processamento e armazenamento. Trabalhando contra tal funcionalidade, no entanto, são dispositivos com recursos limitados, distribuição geográfica de implantações, e inúmeros dispositivos em uma solução.

### Provisionamento e autenticação seguros de dispositivos

O provisionamento seguro de dispositivos significa fornecer uma chave de identidade exclusiva para cada dispositivo, que pode ser usada pela infraestrutura de IoT para se comunicar com o dispositivo enquanto ele estiver em operação. A chave gerada com o ID do dispositivo selecionado pelo usuário forma a base de um token usado em todas as comunicações entre o dispositivo e o Hub de IoT do provedor de nuvem.

Os IDs de dispositivo podem ser associados a um dispositivo durante a fabricação (ou seja, atualizados em um hardware confiável) ou pode usar uma identidade fixa existente como proxy (por exemplo, números de série da CPU). Desde a mudança esta informação de identificação no dispositivo não é simples, é importante introduzir IDs de dispositivos lógicos em caso o hardware do dispositivo subjacente seja alterado, mas o dispositivo lógico permaneça o mesmo. Em alguns casos, a associação de uma identidade de dispositivo pode acontecer no momento da implantação do dispositivo (por exemplo, um engenheiro de campo configura fisicamente um novo dispositivo enquanto se comunica com o back-end da solução).

O registro de identidade do Hub de IoT do provedor de nuvem fornece armazenamento seguro de identidades de dispositivos e chaves de segurança para uma solução. Indivíduos ou grupos de identidades de dispositivos podem ser adicionados a uma lista de permissões, ou a uma lista de bloqueio, permitindo controle total sobre o acesso ao dispositivo.

As políticas de controle de acesso do Hub IoT permitem a ativação e a desativação de qualquer identidade de dispositivo, fornecendo uma

maneira de desassociar um dispositivo de uma implantação de IoT quando necessário. Esta associação e a desassociação de dispositivos é baseada na identidade de cada dispositivo.

Outros recursos de segurança do dispositivo incluem:

- Os dispositivos não aceitam conexões de rede não solicitadas. Eles estabelecem todas as conexões e rotas em uma moda apenas de saída. Para um dispositivo receber um comando do back-end, o dispositivo deve iniciar uma conexão para verificar se há comandos pendentes a serem processados. Uma vez que uma conexão entre, o dispositivo e o Hub IoT são estabelecidos com segurança, mensagens da nuvem para o dispositivo e do dispositivo para a nuvem podem ser enviadas de forma transparente.
- Os dispositivos apenas se conectam ou estabelecem rotas para serviços conhecidos com os quais estão emparelhados, como, por exemplo, IoT Hub do Azure, IoT Core Google, AWS IoT Device Management.
- A autorização e autenticação no nível do sistema usam identidades por dispositivo, tornando as credenciais de acesso e permissões revogáveis quase instantaneamente.

### Conectividade segura

O IoT Hub do Azure, IoT Core Google, AWS IoT Device Management suportam conectividade segura usando protocolos comprovados pelo setor - HTTPS, AMQP e MQTT.

Eles oferecem durabilidade de mensagens entre a nuvem e os dispositivos por meio de um sistema de reconhecimento em resposta às mensagens. A durabilidade extra para mensagens é alcançada armazenando mensagens em cache no Hub IoT por até sete dias para telemetria e dois

dias para comandos. Esta abordagem permite dispositivos que se conectam esporadicamente, devido a problemas de energia ou conectividade, para receber esses comandos.

O Hub IoT do Azure mantém uma fila por dispositivo para cada dispositivo. A escalabilidade requer a capacidade de interoperar com segurança com uma ampla variedade de dispositivos. Hub IoT do Azure permite conexão segura a dispositivos habilitados para IP e não habilitados para IP (usando um dispositivo IoT Edge como Porta de entrada).

Outros recursos de segurança de conexão incluem:

- O caminho de comunicação entre dispositivos e o Hub IoT do Azure, ou entre gateways e o Azure IoT Hub, é protegido usando Transport Layer Security (TLS) padrão do setor com o Hub IoT do Azure autenticado usando o protocolo X.509.
- Para proteger os dispositivos de conexões de entrada não solicitadas, o Hub IoT do Azure não abre nenhuma conexão com o dispositivo. O dispositivo inicia todas as conexões.
- O Hub IoT do Azure armazena mensagens para dispositivos de forma durável e aguarda a conexão do dispositivo. Esses comandos são armazenados por dois dias, permitindo que dispositivos conectados esporadicamente, devido a problemas de energia ou conectividade, recebam esses comandos. O Hub IoT do Azure mantém uma fila por dispositivo para cada dispositivo.

### Práticas recomendadas de segurança de IoT

Proteger uma infraestrutura de Internet das Coisas (IoT) requer uma estratégia rigorosa de segurança em profundidade. Esta estratégia exige que você proteja os dados na nuvem, proteja a integridade dos dados enquanto estiver em trânsito pela rede pública de Internet e dispositivos de

provisionamento seguros. Cada camada cria maior garantia de segurança à infraestrutura.

### Conceitos básicos sobre IoT Hub

A Internet das Coisas (IoT) é normalmente definida como uma rede de dispositivos físicos que se conectam e trocam mensagens com outros dispositivos e serviços pela Internet ou alguma outra rede de comunicação. Existem atualmente mais de dez bilhões de dispositivos conectados no mundo e mais são adicionados a cada ano.

A definição mais clara para IoT é toda e qualquer coisa que possa ser incorporado sensores e software, que possa se conectar pela Internet.

As seguintes tecnologias tornaram a IoT possível:

- Acesso a sensores de baixo custo e baixa potência.
- Vários protocolos que permitem a conectividade com a Internet.
- Plataformas de computação em nuvem, como Azure.
- Big data.
- Aprendizado de máquina (Machine Learning).
- Inteligência artificial.
- Hub IoT do Azure.

O Azure IoT Hub é um serviço gerenciado hospedado na nuvem que atua como um hub de mensagem central para a comunicação entre um aplicativo IoT e seus dispositivos anexados. Você pode conectar milhões de dispositivos e suas soluções de back-end de maneira confiável e segura. Quase qualquer dispositivo pode ser conectado a um Hub IoT.

Ele suporta vários padrões de mensagens, incluindo telemetria de dispositivo para nuvem, upload de arquivos de dispositivos como câmeras de vigilância. O Hub IoT também oferece suporte ao monitoramento para ajudá-lo a rastrear a criação do dispositivo, as conexões do dispositivo e ajuda na gestão de falhas do dispositivo.

O IoT Hub é dimensionado para milhões de dispositivos conectados simultaneamente e milhões de eventos por segundo para suportar cenários bem complexos.

É possível integrar o Hub IoT com outros serviços do Azure para criar soluções completas de ponta a ponta. Por exemplo:

- O Event Grid permite que sua empresa reaja rapidamente a eventos críticos de maneira confiável, escalonável e segura.
- Usando Logic Apps para automatizar processos de negócios.
- Azure Machine Learning para adicionar aprendizado de máquina e modelos de IA à sua solução.
- Azure Stream Analytics para executar cálculos analíticos em tempo real no streaming de dados de seus dispositivos.

O Hub IoT tem um contrato de nível de serviço de 99,9% para o Hub IoT. O SLA completo do Azure explica a disponibilidade garantida do Azure na totalidade.

### AWS IoT Core

O AWS IoT Core oferece suporte ao *Message Queuing Telemetry Transport* (MQTT), um protocolo de mensagens leve amplamente adotado e projetado para dispositivos restritos. Os participantes do MQTT recebem informações organizadas por meio de tópicos do MQTT. Um tópico MQTT

atua como um mecanismo de correspondência entre publicadores e assinantes. Conceitualmente, um tópico MQTT se comporta como um canal de notificação efêmero.

Para AWS IoT, uma das primeiras considerações ao usar o MQTT é a estratégia de design de seus tópicos do MQTT. Os tópicos do MQTT devem equilibrar as comunicações atuais do dispositivo, as operações do lado da nuvem e os recursos futuros do dispositivo. Portanto, pode ser um desafio projetar uma estrutura de tópicos MQTT ideal que crie um esquema suficiente para impor a comunicação de privilégios mínimos, mas não crie uma estrutura rígida que dificulte suportar futuras implantações de dispositivos.

Esse documento fornece as melhores práticas e orientações de design de tópicos MQTT. Ele descreve um conjunto de estruturas de tópicos MQTT comumente usadas que podem ser implementadas para resolver vários padrões de mensagens de dispositivos e, em seguida, aplica vários padrões de design de exemplo usando diferentes serviços AWS IoT.

### Padrões de comunicação MQTT

Os aplicativos de IoT oferecem suporte a vários cenários de comunicação, como dispositivo para dispositivo, dispositivo para nuvem, nuvem para dispositivo e dispositivo para usuários. Embora a gama de padrões possa variar significativamente, a maioria dos modelos de comunicação MQTT deriva de três padrões MQTT: ponto a ponto, broadcast e fan-in.

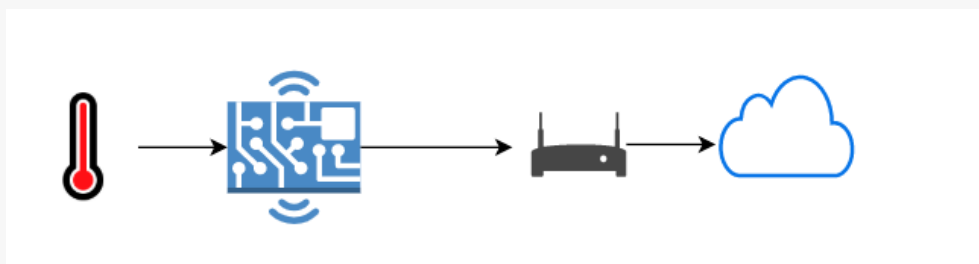
#### Ponto a ponto

Um padrão de comunicação ponto a ponto é um dos blocos de construção básicos de como os dispositivos geralmente enviam e recebem mensagens no MQTT. Dois dispositivos usam um único tópico MQTT como canal de comunicação. O dispositivo que recebe o evento assina um tópico MQTT. A coisa que envia a mensagem publica no mesmo tópico MQTT



conhecido. Essa abordagem é comum em cenários de casa inteligente em que um usuário final recebe atualizações sobre a coisa em casa. No exemplo a seguir, a ocupação da sala publica uma mensagem sobre um tópico inscrito por um aplicativo executado no display digital fora da sala de projeção.

Figura 5 - Comunicação ponto a ponto.

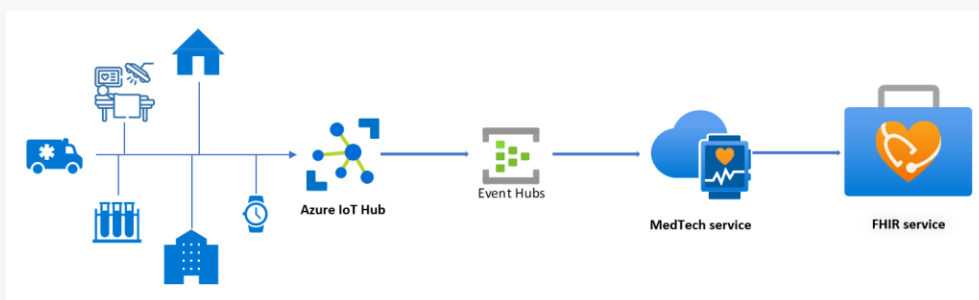


Fonte: <https://eximia.co/use-azure-functions-para-ingestao-de-dados-em-seu-projeto-de-iot/>.

A comunicação ponto a ponto não se limita à comunicação um a um entre dispositivos. O ponto a ponto também é usado na comunicação um para muitos, onde um único editor pode publicar em dispositivos individuais usando um tópico MQTT diferente por dispositivo.

Essa abordagem é comum em cenários de notificação em que um administrador envia atualizações distintas para dispositivos específicos. No exemplo a seguir, o serviço de reparo usa um conjunto de comunicações ponto a ponto para fazer um loop programático por meio de uma lista de appliances e publicar uma mensagem.

Figura 6 - Mensagem de um para muitos dispositivos.



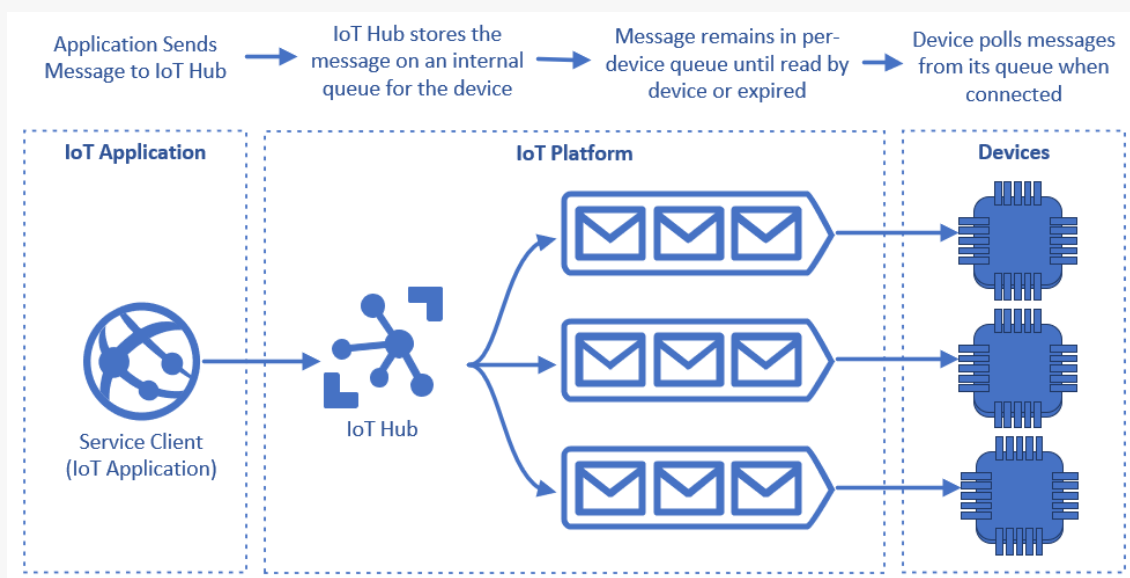
Fonte: <https://learn.microsoft.com/pt-br/azure/healthcare-apis/iot/device-messages-through-iot-hub>.

## Broadcast

Os padrões de transmissão são usados para mensagens de um para muitos. O padrão de transmissão envia a mesma mensagem para uma grande frota de dispositivos. Em uma transmissão, vários dispositivos assinam o mesmo tópico MQTT e o remetente publica uma mensagem nesse tópico. Um uso típico de um padrão de transmissão é enviar uma notificação para dispositivos com base na categoria ou grupo do dispositivo. Por exemplo, uma estação meteorológica transmite uma mensagem de transmissão com base em um tópico com base em sua geolocalização.

A ilustração a seguir mostra um exemplo em que um padrão de transmissão envia uma mensagem sobre um tópico de clima que todos os veículos de entrega no estado assinam. A mensagem inclui condições meteorológicas e coordenadas detalhadas de localização. Com base na localização atual do veículo, ele pode ignorar a mensagem ou realizar alguma ação.

Figura 7 - Mensagens um-para-muitos na comunicação de transmissão.



Fonte: <https://learn.microsoft.com/pt-br/azure/architecture/example-scenario/iot/cloud-to-device>.

## Introdução ao IoT Central

### O que é Azure IoT Central?

IoT Central é uma plataforma de aplicativo IoT que reduz a carga de trabalho e o custo de desenvolvimento e gerenciamento, mantendo soluções de IoT ao nível empresarial. Optar por construir com o IoT Central dá a você a oportunidade de concentrar tempo, dinheiro e energia na transformação de seus negócios com dados de IoT, em vez de apenas manter e atualizar uma infraestrutura de IoT complexa e em constante evolução.

A IU da web permite monitorar as condições do dispositivo, criar regras e gerenciar milhões de dispositivos e seus dados ao longo de seu ciclo de vida. Além disso, permite que você atue conforme os insights do dispositivo, estendendo IoT a inteligência em aplicativos de linha de negócios.

Como um desenvolvedor em IoT, é importante saber quando é apropriado escolher o Azure IoT Central para o seu negócio ou para o cliente e como implementar esta alternativa fácil de gerenciar ao máximo em opção de solução PaaS. Também é importante reconhecer que os recursos de IoT Plug and Play no Azure IoT Central que atualmente em versão pública de preview.

Para ambientes de produção, use um aplicativo central IoT criado a partir de um modelo de aplicativo atual e geralmente disponível.



**XP**e

# > Capítulo 7



## Capítulo 7. Azure Power Platform

---

As empresas mais modernas funcionam com base em dados. Os usuários interagem com os dados diariamente ao inserir seu tempo na folha de pagamento, buscar orientação sobre os processos existentes e analisar dados para tomar decisões. Atualmente a tecnologia orienta o mundo todo, os usuários podem ter autonomia para obter insights e interagir com os dados, ao mesmo tempo em que automatizam responsabilidades que parecem mais um fardo do que a tarefa do trabalho. A Power Platform viabiliza o seu negócio para criar soluções e, ao mesmo tempo, capacitá-lo a unir tecnologia personalizada para ajudar a todos, desde o CEO aos trabalhadores da linha de frente, auxiliando na condução dos negócios com dados.

### Power Apps

Power Apps fornece um ambiente de desenvolvimento rápido de baixo código para a construção de aplicativos personalizados para as necessidades de negócios. Ele tem serviços, conectores e um serviço de dados escalável e plataforma de aplicativos para permitir integração e interação simples com os dados existentes. Power Apps permite a criação de aplicativos web e móveis que rodam em todos os dispositivos. As pessoas usam aplicativos para todas as áreas de suas vidas, e os negócios não devem ser exceção. A maioria das soluções prontas para o uso não atende às necessidades comerciais exatas ou se integra bem a outros programas comerciais. Power Apps facilita o desenvolvimento de aplicativos com uma interface simples para que todo usuário de negócios ou desenvolvedor profissional possa construir aplicativos personalizados.

## Power Automate

O Power Automate permite que os usuários criem fluxos de trabalho automatizados entre aplicativos e serviços. Isso ajuda automatizar processos de negócios repetitivos, como comunicação, coleta de dados e aprovações de decisões.

Não perca horas produtivas importantes redigindo o mesmo e-mail para uma atualização semanal ou caminhando aprovações. Não apenas para o usuário individual, o Power Automate permite a criação de automação de processos de nível empresarial. A interface simples do Power Automate permite que todos os níveis de usuário automatizem as tarefas de trabalho, de iniciantes a desenvolvedores experientes.

## Power BI

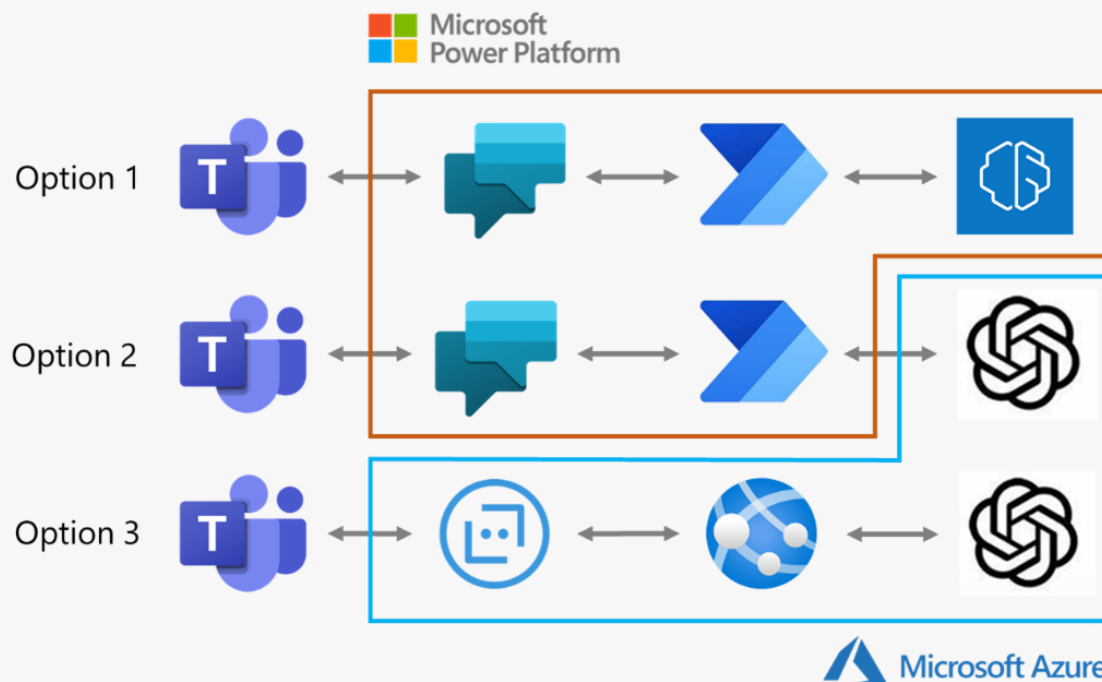
Power BI (Business Intelligence) é um serviço analítico para negócios que fornece insights para análise de dados. Ele pode compartilhar esses insights por meio de visualizações de dados que compõem relatórios e painéis para permitir decisões rápidas e informadas. O Power BI pode ser escalado em uma organização com governança e segurança integradas, permitindo que as empresas se concentrem no uso de dados mais do que em gerenciá-los. Você pode considerar o Power BI como a perna de análise e percepções da Power Platform. Ele pega dados de negócios e permite que você os exiba da maneira que faz mais sentido para os usuários. Um painel do Power BI poderia potencialmente substituir uma reunião permanente para relatar as métricas da empresa, como dados de vendas, progresso em relação às metas ou desempenho do funcionário.

## Power Virtual Agents

Power Virtual Agents permite que qualquer pessoa crie chatbots poderosos usando um gráfico guiado e sem interface código, sem a necessidade de cientistas de dados ou desenvolvedores.

O Power Virtual Agents aborda muitos dos principais problemas com a construção de chatbot. Elimina a lacuna entre especialistas no assunto e as equipes de desenvolvimento que criam os chatbots. Remove a complexidade de expor às equipes IA conversacional e à necessidade de escrever códigos complexos. Isto minimiza o esforço de TI necessário para implantar e manter uma solução de conversação personalizada ao capacitar especialistas no assunto para construir e manter suas próprias soluções de conversação.

## Exemplo de Solução



Os usuários ou equipes (representados pelos ícones 'Teams') interagem ou utilizam uma solução de comunicação, ou colaboração Power Virtual Agent. Esta interação pode acionar processos ou fluxos de trabalho

no Power Automate, que por sua vez pode integrar-se com outras ferramentas, como LUIS, nesse caso representado pelo ícone do Azure Cognitive Services.

No primeiro e no segundo fluxo, após a interação inicial e o processamento pelo Power Automate, a informação ou o comando é enviado ao Azure Logic Apps, que pode gerenciar tarefas mais complexas de integração ou automação.

Já no terceiro fluxo, após a ação no Power Automate, parece que há algum tipo de processo de automação representado pelo Bot Service. Este processo, por sua vez, também se conecta ao Azure Logic Apps.

Em resumo, toda essa ilustração demonstra que o Power Virtual Agent tem canais, assim como o Bot Service, para integrar facilmente com outras soluções e atuar como um serviço intermediário para troca de mensagens e atuar como agentes de conversação em todos os níveis.





**XP**e

# > Capítulo 8



## Capítulo 8. Ferramentas para Gerenciamento e Configuração de Ambientes no Azure

---

### Azure Portal

O portal é o recurso oficial para gerenciar seus recursos no Azure, disponível no endereço <https://portal.azure.com> ou <https://preview.portal.azure.com>, para obter todos os recursos relacionados à interface, ainda em testes no portal.

### Azure Mobile App

Azure Mobile App é uma aplicação móvel, disponível para Android e iOS, onde podemos monitorar as assinaturas e recursos existentes em nossa infraestrutura, assim como acessar o Azure Cloud Shell, um terminal on-line para gerenciar os recursos.

### Azure PowerShell

O Azure PowerShell é um conjunto de comandos cmdlets para gerenciar recursos do Azure diretamente da linha de comando via PowerShell. O design do Azure PowerShell facilita o aprendizado para quem começa a usar, mas conta com recursos avançados de automação. Escrito em .NET Standard, o Azure PowerShell funciona com o PowerShell 5.1 no Windows e com o PowerShell 7.x e versões posteriores em todas as plataformas.

### Azure CLI

A CLI do Azure (interface de linha de comando do Azure) é um conjunto de comandos usado para criar e gerenciar recursos do Azure. A CLI do Azure está disponível nos serviços do Azure e foi criada para ajudar você a trabalhar de maneira rápida com o Azure, com ênfase na automação.

## Cloud Shell

O Azure Cloud Shell é um shell interativo, autenticado e acessível pelo navegador para o gerenciamento de recursos do Azure. Link direto para utilizar o Azure Cloud Shell <https://shell.azure.com>.

## Templates ARM

Com a mudança para a nuvem, muitas equipes adotaram métodos de desenvolvimento ágeis. Essas equipes iteram rapidamente. Elas precisam implantar repetidamente suas soluções na nuvem e saber que sua infraestrutura está em um estado confiável. À medida que a infraestrutura se tornou parte do processo iterativo, a divisão entre operações e desenvolvimento desapareceu. As equipes precisam gerenciar a infraestrutura e o código do aplicativo por meio de um processo unificado.

Para atender a esses desafios, você pode automatizar implantações e usar a prática da infraestrutura como código. No código, você define a infraestrutura que precisa ser implantada. O código de infraestrutura torna-se parte do seu projeto. Assim como o código do aplicativo, você armazena o código de infraestrutura em um repositório de origem e controla a versão dele. Qualquer um em sua equipe pode executá-lo e implantar ambientes semelhantes em minutos.



**XP**e

# > Capítulo 9

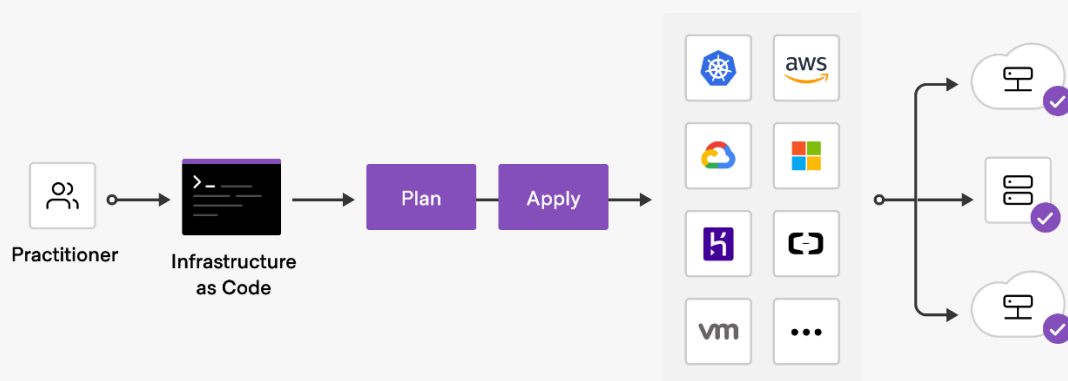


## Capítulo 9. Soluções de Infraestrutura como Código (IaC)

### O que é a iac (infraestrutura como código)?

A IaC (infraestrutura como código) usa a metodologia e o controle de versão do DevOps com um modelo descritivo para definir e implantar a infraestrutura, como redes, máquinas virtuais, balanceadores de carga e topologias de conexão. Assim como o mesmo código-fonte sempre gera o mesmo binário, um modelo de IaC gera o mesmo ambiente sempre que ele é implantado.

Figura 8 – Diagrama de Infra as a Code.



Fonte: [developer.hashicorp.com](https://developer.hashicorp.com).

IaC é uma prática de DevOps chave e um componente de [entrega contínua](#). Com a IaC, as equipes do DevOps podem trabalhar juntamente com um conjunto unificado de práticas e ferramentas para fornecer aplicativos e sua infraestrutura de suporte de forma rápida e confiável em escala.

### Evitar a configuração manual para impor consistência

A IaC evoluiu para resolver o problema de *descompasso de ambiente* nos pipelines de lançamento. Sem IaC, as equipes devem manter as configurações de ambiente de implantação individualmente. Com o

tempo, cada ambiente se torna um "flocos de neve", uma configuração exclusiva que não pode ser reproduzida automaticamente. A inconsistência entre ambientes pode causar problemas de implantação. A administração e a manutenção da infraestrutura envolvem processos manuais propensos a erros e difíceis de rastrear.

A IaC evita a configuração manual e impõe a consistência representando os estados de ambiente desejados por meio de código bem documentado em formatos como JSON. As implantações de infraestrutura com IaC são repetíveis e evitam problemas de runtime causados por descompasso de configuração ou dependências ausentes. Os pipelines de versão executam as descrições de ambiente e os modelos de configuração de versão para configurar ambientes de destino. Para fazer alterações, a equipe edita a origem, não o destino.

A *Idempotência* – capacidade de uma determinada operação de sempre produzir o mesmo resultado – é um princípio de IaC importante. Um comando de implantação sempre define o ambiente de destino na mesma configuração, independentemente do estado inicial do ambiente. A Idempotência é obtida configurando automaticamente o destino existente ou descartando o destino existente e recriando um novo ambiente.

### Fornecer ambientes de teste estáveis rapidamente em escala

A IaC ajuda as equipes do DevOps a testar aplicativos em ambientes semelhantes à produção no início do ciclo de desenvolvimento. O Teams pode provisionar vários ambientes de teste de forma confiável sob demanda. A nuvem provisiona e rasga ambientes dinamicamente com base nas definições de IaC. O próprio código de infraestrutura pode ser validado e testado para evitar problemas comuns de implantação.

## Usar arquivos de definição declarativa

A IaC deve usar arquivos de definição declarativa, se possível. Um arquivo de definição descreve os componentes e a configuração que um ambiente requer, mas não necessariamente como alcançar essa configuração. Por exemplo, o arquivo pode definir uma versão e uma configuração necessárias do servidor, mas não especificar o processo de instalação e configuração do servidor. Essa abstração permite maior flexibilidade para usar técnicas otimizadas fornecidas pelo provedor de infraestrutura. As definições declarativas também ajudam a reduzir a dívida técnica de manutenção do código imperativo, como scripts de implantação, que podem se acumular ao longo do tempo.

Não há sintaxe padrão para IaC declarativa. A sintaxe para descrever IaC geralmente depende dos requisitos da plataforma de destino. Diferentes plataformas dão suporte a formatos de arquivo, como YAML, JSON e XML, vamos abordar duas plataformas durante nosso curso, Ansible e Terraform, abaixo vamos descrever um pouco sobre cada uma das plataformas.

## O que é Ansible

Ansible é uma plataforma open source (código aberto) da Red Hat que automatiza o provisionamento de recursos em nuvem, o gerenciamento de configuração e a implantação de aplicativos. Usando o Ansible, você pode provisionar VMs, contêineres e todo sua infraestrutura em nuvem. Além de provisionar e configurar aplicativos e seus ambientes.

O Ansible permite que você automatize a implantação e configuração de recursos em seu ambiente, como redes virtuais, armazenamento, sub-redes e grupos de recursos.

Ele foi projetado para implantações de vários níveis, o Ansible não tem agente, o que significa que você não precisa instalar software nas máquinas gerenciadas.

Ansible também modela sua infraestrutura de TI, descrevendo como todos os seus sistemas se inter-relacionam, ao invés de gerenciar apenas um sistema de cada vez.

### O que é Terraform

Terraform é uma ferramenta de open source (código aberto) que permite provisionar, gerenciar e criar versões em nuvem da sua infraestrutura. Ele codifica a infraestrutura em arquivos de configuração que descrevem a topologia dos recursos de nuvem, como VMs, contas de armazenamento (storages) e interfaces de rede.

A interface via linha de comando (CLI) do Terraform fornece um mecanismo simples para implantar e versão dos arquivos de configuração para o Azure ou qualquer outro serviço de nuvem com suporte. A CLI também permite que você valide e visualize as mudanças na infraestrutura antes de implementá-las.

O Terraform também oferece suporte a cenários de várias nuvens. Isso significa que permite que os desenvolvedores usem as mesmas ferramentas e arquivos de configuração para gerenciar a infraestrutura em vários provedores de nuvem.

Você pode executar o Terraform interativamente a partir da CLI com comandos individuais ou de forma não interativa como parte de um pipeline de integração contínua.

Também existe uma versão empresarial do Terraform disponível, Terraform Enterprise.





**XP**e

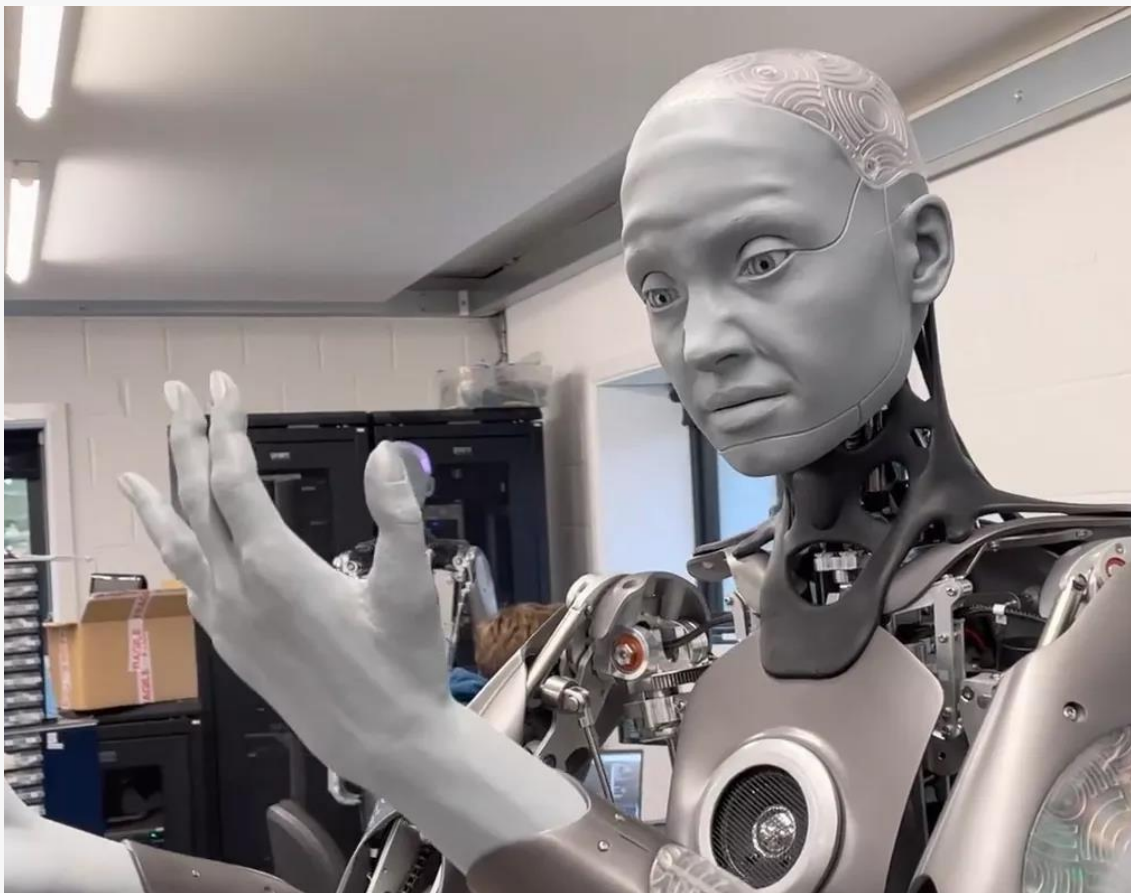
# > Capítulo 10



## Capítulo 10. Inteligência Artificial (IA) e Machine Learning

Não é novidade que já temos robôs humanoides sendo criados, como, por exemplo, o projeto Ameca, da empresa britânica *Engineered Arts*, especializada em máquinas humanoides. Sua semelhança com um humano é impressionante.

Figura 9 – Modelo AMECA robô humanoide.



Fonte: [g1.globo.com](http://g1.globo.com).

Assustador ou não, já existe, e pode ser alugado ou comprado, além de ser possível programá-lo para fazer o que for necessário.

Trouxe esse case para falarmos de Inteligência Artificial e Machine Learning, duas tecnologias que também já estão presentes no nosso cotidiano.

- Você conhece essas tecnologias?

Inteligência Artificial é uma das áreas da Ciência da Computação que faz com que computadores e máquinas desenvolvam a capacidade de aprender e compreender as coisas, semelhante ao cérebro humano, agindo de forma independente. Já o Machine Learning é uma espécie de subconjunto da IA.

A ideia é se aproximar ao máximo ao sistema cognitivo humano, compreender, adaptar e dar sentido aos dados absorvidos, como um texto lido, algo ouvido, uma imagem visualizada, vídeos etc. Se aproximar a toda forma de comunicação como nós humanos interagimos.

- Uma coisa é uma coisa, outra coisa é outra coisa

Muito se fala por aí sobre Inteligência Artificial e Machine Learning, mas, afinal, o que são essas tecnologias de fato e quais suas diferenças?

Bom, em linhas gerais, Inteligência Artificial é basicamente transpor a inteligência humana em uma máquina imitando o seu cérebro. Já Machine Learning é um método de aprendizagem que, por meio de coleta de dados, torna uma máquina capaz de prever resultados diversos ou entender padrões.

## O Algoritmo

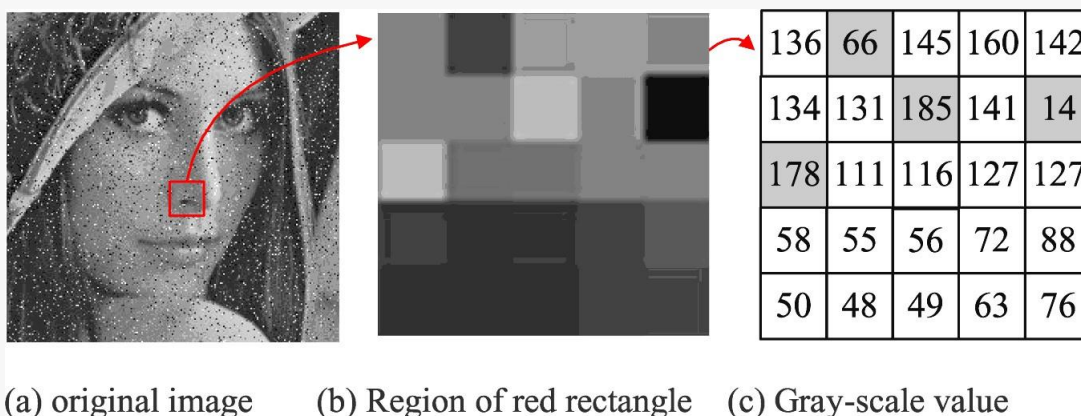
### As máquinas aprendem do nada?

A resposta é não. Existe um algoritmo extremamente complexo e "inteligente" por trás de tudo.

No aprendizado de máquina, o algoritmo instrui o computador no que ele deve fazer para compreender os padrões e assim ter o resultado desejado.

O princípio do funcionamento de um algoritmo de uma IA é converter informações em números, porque, na prática, este algoritmo é um cálculo de probabilidades.

Figura 10 – Imagem pixelada convertida para números.



Fonte: An Efficient Denoising Architecture for Removal of Impulse Noise in Images.

### Alimente-me e será seu fim

No aprendizado de máquina, o algoritmo instrui o computador no que ele deve fazer para compreender os padrões e assim ter o resultado desejado. Quanto mais dados o algoritmo é exposto, mais ele aprende e melhora seu desempenho nos resultados, isso é fato. Como o algoritmo trabalha com probabilidades, ele precisa de muitos dados para executar uma função matemática para compreender e encontrar um padrão.

Desenvolvendo sua própria inteligência ao longo do tempo, inclusive usando um recurso próprio que se chama BackPropagation, que é um método de aprendizagem, a IA vai aprender com seus próprios erros. O erro servirá para melhorar os parâmetros da rede neural para as próximas respostas. Lembra da Skynet e Hall 9000? Dois sucessos do cinema protagonizaram uma revolução das máquinas, aonde essas IA's se tornaram autoconscientes e criaram suas próprias regras.

## Ciência de dados e Deep Learning

### Mostre-me os números

Um subconjunto da Inteligência Artificial, a Data Science fica com a parte burocrática da coisa, que envolve estatística e análise de dados.

O cientista de dados, por exemplo, é que transforma esses dados em informação útil e estratégica. No Machine Learning ele é figura indispensável para otimizar os resultados, identificando tendências e estruturando grandes massas de dados, transformando-os em informação útil.

### Águas profundas da aprendizagem

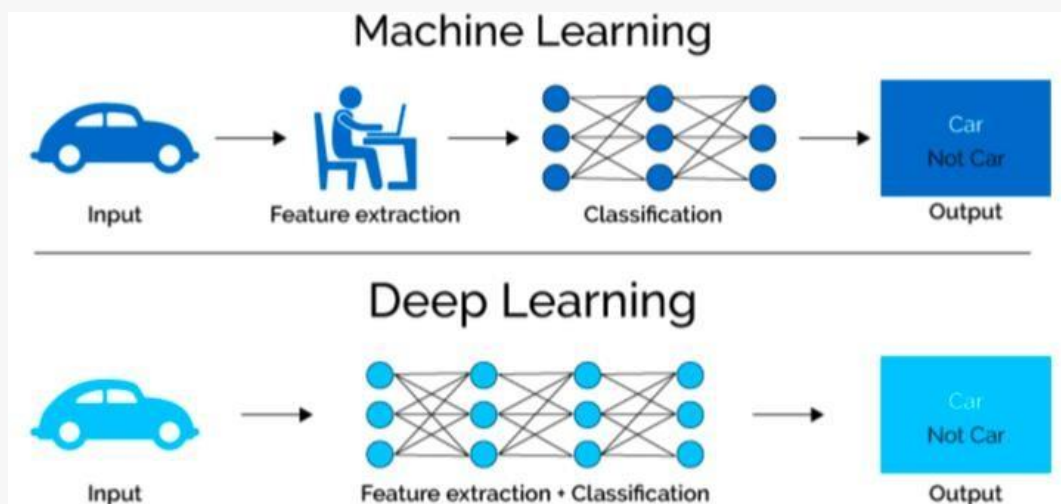
Deep Learning é um subconjunto do Machine Learning e permite os computadores resolverem problemas mais complexos sem a intervenção humana.

O termo refere-se a redes neurais artificiais capazes de tomar decisões e aprender com um altíssimo nível de precisão.

Deep Learning permitiu aplicações práticas de Machine Learning e IA. Carros que dirigem sozinhos, saúde preventiva e até mesmo

recomendação de filmes, músicas e séries. Com a ajuda de Deep Learning, IA pode chegar próximo a um estado semelhante à ficção científica.

Figura 11 – Machine Learning vs. Deep Learning.



Fonte: Slide da palestra do Pierre Guillou.



**XP**e

# > Capítulo 11



## Capítulo 11. Machine Learning com Plataformas em Nuvem

---

### Conceitos básicos de Inteligência Artificial

Machine learning é a base para a maioria das soluções de inteligência artificial e é onde começa a criação de uma solução inteligente para treinar um modelo de previsão usando os dados históricos que você coletou.

O Azure Machine Learning e o Amazon SageMaker são serviços de nuvem que você pode usar para treinar e gerenciar modelos de machine learning.

As ferramentas de machine learning em nuvem fornecem uma plataforma para operacionalizar as cargas de trabalho necessárias para conduzir o processo iterativo que permite a entrega de aplicativos de aprendizado de máquina baseados em dados.

Existem três principais tipos de usuário que as plataformas de machine learning em nuvem oferecem suporte neste processo:

- Cientistas de dados, que usam seus conhecimentos de estatística e análise de dados para conduzir experimentos analíticos e treinar modelos de aprendizado de máquina. Esses usuários geralmente trabalham em Python ou R e usam estruturas como Scikit-Learn, PyTorch e TensorFlow para treinar modelos de aprendizado de máquina.
- Cientistas de dados “Citizen” (cientista de dados especializados em matemática e estatística) e desenvolvedores de aplicativos, que trabalham principalmente no campo de dados estatísticos e análises estatísticas, mas que precisam treinar modelos de aprendizado de máquina para dar suporte a aplicativos. Esses usuários podem tomar



vantagem das ferramentas gráficas que abstraem a complexidade subjacente do treinamento do modelo.

- Engenheiros e operadores de software, que precisam operacionalizar o aprendizado de máquina para dar suporte a aplicativos e serviços. Suas tarefas envolvem normalmente o uso de scripts ou processos automatizados de DevOps para gerenciar o retreinamento e a implantação do modelo, bem como o monitoramento geral de aplicativos e a solução de problemas.

### O que é o aprendizado de máquina?

Machine learning é uma técnica que usa matemática e estatística para criar um modelo que pode prever valores desconhecidos.

Por exemplo, suponha que *Adventure Works Cycles* é uma empresa que aluga bicicletas em uma cidade. A empresa poderia usar dados históricos para treinar um modelo que prevê a demanda diária de aluguel a fim de garantir que haja funcionários e bicicletas suficientes disponíveis.

Para fazer isso, a Adventure Works poderia criar um modelo de machine learning que extrai informações sobre um dia específico (o dia da semana, as condições meteorológicas antecipadas e assim por diante) como uma entrada e prevê o número esperado de aluguéis como uma saída.

Matematicamente, você pode considerar o machine learning como uma forma de definir uma função (vamos chamá-la de  $f$ ) que opera em uma ou mais *funcionalidades* de algo (que chamaremos de  $x$ ) para calcular um *rótulo* previsto ( $y$ ), desta forma:

$$f(x) = y$$

Neste exemplo de aluguel de bicicletas, os detalhes sobre determinado dia (dia da semana, clima etc.) são as *funcionalidades* ( $x$ ), o número de aluguéis desse dia é o *rótulo* ( $y$ ) e a função ( $f$ ) que calcula o número de aluguéis com base nas informações sobre o dia é encapsulada em um modelo de machine learning.

A operação específica que a função  $f$  executa em  $x$  para calcular  $y$  depende de vários fatores, incluindo o tipo de modelo que você está tentando criar e o algoritmo específico usado para treinar o modelo. Além disso, geralmente, os dados usados para treinar o modelo de machine learning exigem algum pré-processamento antes que o treinamento do modelo possa ser executado.

### Azure Machine Learning

A Microsoft apostou tudo no [Machine Learning Studio](#), uma ferramenta sofisticada e gráfica para encontrar sinais em seus dados. É como uma planilha para IA. Há uma interface do tipo drag-and-drop para criação de fluxogramas que ajudam a entender seus números.

A documentação diz que “não é necessário codificação” e isso é tecnicamente verdade, mas você ainda precisa pensar como um programador para usá-la efetivamente. Você simplesmente não ficará tão atolado na estruturação de seu código. E se perder os erros de sintaxe, a digitação dos dados, e as outras alegrias da programação, poderá importar módulos escritos em Python, R ou várias outras opções.

A opção mais interessante é que a Microsoft adicionou a infraestrutura para aproveitar o que você aprende com a Inteligência Artificial e transformar o modelo preditivo em um serviço Web em execução no Azure. Assim, você cria seu conjunto de treinamento, cria seu modelo e,

em apenas alguns cliques, fornece respostas em pacotes JSON do seu serviço Azure.

### Amazon SageMaker

A Amazon criou o [SageMaker](#) para simplificar o uso de suas ferramentas de machine learning. O Amazon SageMaker une as diferentes opções de armazenamento da AWS (S3, Dynamo, Redshift etc.) e envia os dados para contêineres do Docker que executam as bibliotecas populares de machine learning (TensorFlow, MXNet, Chainer etc.). Todo o trabalho pode ser rastreado com os notebooks Jupyter antes que os modelos finais sejam implantados como APIs próprias. O SageMaker move seus dados para as máquinas da Amazon para que você possa se concentrar em pensar nos algoritmos e não no processo. Se você deseja executar os algoritmos localmente, sempre é possível fazer o download das imagens do Docker para simplificar.

### DataRobot

Muitas das abordagens aqui permitem criar um modelo de machine learning em um clique. O DataRobot possui a capacidade de construir centenas de modelos simultaneamente, também com apenas um clique. Quando os modelos são feitos, você pode escolher entre eles e descobrir qual faz a melhor previsão e seguir com isso. O segredo é um “mecanismo de processamento massivamente paralelo”, em outras palavras, uma nuvem de máquinas fazendo a análise.

O DataRobot está expandindo através da implementação de novos algoritmos. A empresa adquiriu recentemente a Nutonian, cujo mecanismo Eureqa deve melhorar a capacidade da plataforma automatizada de criar

séries temporais e modelos de classificação. O sistema também oferece uma API do Python para usuários mais avançados.

Ele está disponível por meio do DataRobot Cloud ou de uma edição de software corporativo que vem com um engenheiro embarcado.

### Google Cloud Machine Learning Engine

O Google investiu pesadamente no TensorFlow, uma das bibliotecas padrão de código aberto para encontrar sinais em dados, e agora você pode experimentar o TensorFlow na nuvem do Google. Algumas das ferramentas do [Google Cloud Machine Learning Engine](#) são de código aberto e essencialmente gratuitas para qualquer pessoa que queira baixá-las, e outras fazem parte das opções comerciais da Google Cloud Platform. Isso dá a você a liberdade de explorar e evitar algum bloqueio porque grande parte do [código é aberto](#) e mais ou menos pronto para ser executado em Mac, Windows ou Linux.

O lugar mais fácil para começar pode ser o [Colaboratory](#), que conecta o Jupyter Notebook com o back-end TensorFlow do Google, para que você possa esboçar seu código e vê-lo rodar. O Google também oferece o [TensorFlow Research Cloud](#) para cientistas que queiram experimentá-lo. Quando for apropriado, você pode executar seus modelos de aprendizado de máquina no hardware do Google acelerado por GPUs ou TPUs.

### IBM Watson Studio

O Watson abrange grande parte do esforço de Inteligência Artificial da IBM. O [IBM Watson Studio](#) é uma ferramenta para explorar seus dados e modelos de treinamento na nuvem ou on-premise.

Você pode usar a versão baseada em nuvem para estudar seus dados e desfrutar de todo o poder que vem com os recursos elásticos e o repositório centralizado. Ou você pode fazer praticamente o mesmo com a privacidade e a conveniência do seu desktop com firewall.

### Azure OpenAI

Azure OpenAI ou OpenAI? Essa é uma pergunta que muita gente tem feito ultimamente. Se você está confuso, deixe-me explicar para você. Ambos os serviços são derivados da mesma tecnologia OpenAI e, portanto, são construídos na mesma estrutura fundamental. No entanto, eles variam na embalagem de seus produtos, por exemplo, oferecendo diferentes parâmetros de nível de serviço, como confiabilidade e limites de taxa. Você pode achar um mais adequado que o outro.

A OpenAI é proeminente pela sua experiência em investigação de inteligência artificial e pelo seu compromisso com o desenvolvimento de modelos e tecnologias avançadas de IA, particularmente no campo do processamento de linguagem natural baseado em transformadores. A empresa fez avanços significativos com modelos como GPT-4, GPT-3, Codex e DALL-E.

O Azure OpenAI é um esforço colaborativo entre o Microsoft Azure e o OpenAI que reúne os poderosos modelos de IA do OpenAI com a infraestrutura segura e escalonável do Azure. Ao aproveitar a infraestrutura do Azure, o Azure OpenAI oferece acesso de nível empresarial aos modelos do OpenAI, incluindo GPT-4, GPT-3, Codex e DALL-E.

### Para que serve?

- Gerar conteúdo.
- Resumir textos.



- Traduzir linguagem natural para código.
- Explicar imagens.

#### Como eu uso?

Tem APIs REST, SDK em Python e até uma interface Web no Estúdio do Azure OpenAI.