



Aprenda com quem faz

Soluções para Desenvolvimento e Ferramentas de Gerenciamento

Henrique Eduardo Souza

2022



SUMÁRIO

Capítulo 1. Azure DevOps	6
O que é Azure DevOps?	6
Tipos de Assinatura AzureDevOps.....	6
Conhecendo os Recursos	8
Capítulo 2. Controle de Versão.....	11
Benefícios da Utilização de Controle de Versão	11
Introdução ao Azure Repos.....	13
Github x Git Comandos Básicos	13
Como usar Git no universo empresarial	15
Trabalhar com Pull Requests.....	18
Capítulo 3. Azure Pipelines	20
O que é CI e CD	21
Demonstração: implementando Integração Contínua (IC)	22
Demonstração: implementando Entrega Contínua (CD)	22
Capítulo 4. Azure DevTest Labs	24
O que é Azure DevTest Labs	24
Principais Vantagens.....	24
Demonstração: criando seu primeiro ambiente.....	24
Capítulo 5. Ferramentas para Gerenciamento e Configuração de Ambientes no Azure.....	26
Azure Portal	26
Azure Mobile App	26
Azure PowerShell	26
Azure CLI	26
Cloud Shell	27

Templates ARM	27
Capítulo 6. Soluções para Infraestrutura como Código (IaC) no Azure	29
O que é Ansible	29
O que é Terraform	29
Capítulo 7. Azure App Service	32
O que é Azure App Service	32
Quais plataformas podemos trabalhar	32
Camadas de Preço	32
Demonstração: configurando e publicando minha primeira Aplicação Azure App Service	33
Demonstração: configurando Integração Contínua	33
Capítulo 8. Azure Functions	35
O que é Serverless e como funciona	35
Compreendendo os modelos de bind das Azure Functions	35
Demonstração: implementando uma Azure Function via Portal	36
Demonstração: implementando uma Azure Function via Visual Studio	36
Demonstração: implementando uma Azure Function via Visual Studio Code	36
Capítulo 9. Azure Logic Apps	38
O que são Logic Apps	38
Demonstração: criando Logic Apps para receber Tweets	39
Demonstração: criando uma API POST com Logic Apps	39
Demonstração: criando uma API GET com Logic Apps	39
Capítulo 10. Azure IoT Hub	41
Conceitos básicos sobre o IoT Hub	41
Demonstração: trabalhando com emuladores IoT e IoT Hub	42

Demonstração: registrando devices IoT e enviando mensagens via C#	42
Capítulo 11. IoT Central	44
Introdução a IoT Central.....	44
Demonstração: como criar seu primeiro IoT Central	44
Capítulo 12. Azure Sphere.....	46
O que é um Azure Sphere.....	46
Cenários para o uso de Azure Sphere.....	47
Principais funcionalidades	48
Capítulo 13. Azure Power Platform	51
Power Apps	51
Power Automate	51
Power BI	52
Power Virtual Agents.....	52
Referências	54



XPe

> Capítulo 1

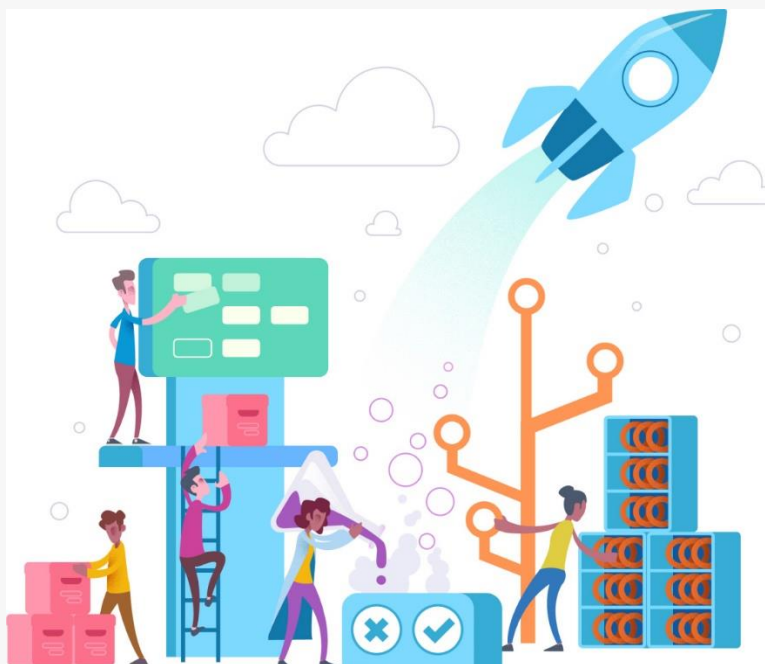


Capítulo 1. Azure DevOps

O que é Azure DevOps?

É uma solução Microsoft para organização de projetos, gerenciamento de atividades, gestão de capacidade, organização de Pipelines CI e CD, que vamos abordar durante o módulo. Também podemos contar com gestão de cenários de testes, gestão de artefatos e um poderoso repositório Git e ou TFS.

Figura 1 – Azure DevOps



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

Tipos de Assinatura AzureDevOps

Esta é uma dúvida frequente entre muitos clientes para os 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 o licenciamento de Microsoft Visual Studio. O licenciamento da Microsoft para Visual Studio Professional custa em torno de R\$ 5.300,00, e é 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, isto é, usuários chave para o negócio, pode-se dizer que eles não gastariam um centavo para trabalhar colocando suas histórias e acompanhando seus projetos. “Mas como assim, Henrique? Você acabou de dizer que uma licença custa R\$ 5.300,00!” Pois bem, demonstrarei a seguir, conforme a Microsoft (2021), os modelos e preços de contratação, que podem facilmente ajudar na decisão da sua companhia.

Plano Básico	Plano Básico + Test Plans
Azure Pipelines: Inclui a oferta gratuita de SERVIÇOS INDIVIDUAIS. Azure Boards: Acompanhamento de item de trabalho e quadros Kanban. Azure Repos: Repositórios Git privados ilimitados. Azure Artifacts: 2 GiB gratuitos por organização.	Inclui todas as funcionalidades do plano Básico. Testar o planejamento, o acompanhamento e a execução. Testes baseados no navegador com anotação. Execução de teste do cliente avançado. Teste de aceitação do usuário. Relatório centralizado.
Primeiros 5 usuários Gratuitos Após R\$ 29,40 por usuário	Média de R\$ 255,00 mês

No entanto, essas informações ainda não são suficientes. Lembre-se do cliente que comentou que havia 300 usuários e que a fábrica de Software 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, normalmente, é por conta do contrato da fábrica de software. Logo, seus 300 usuários podem

utilizar a ferramenta de Boards, organizar toda capacidade do projeto e 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 gerenciarem 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, de forma rápida e fácil, 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 itens de trabalho disponíveis para seu projeto.

Test Plans

O Test Plans é o único serviço do Azure comercializado com uma licença específica e 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 que são criados no Test Plans. Nesse sentido, com a união de todos os recursos, podemos executar, acompanhar, rastrear e criar métricas de qualidade do software baseando-se em todo o ciclo de vida de seu desenvolvimento.

Artifacts

O Azure Artifacts é uma maneira simples e segura de manter as dependências do seu projeto de software. Quem nunca teve problemas ao

iniciar um projeto de um novo cliente e havia dezenas de pacotes espalhados na máquina de algum desenvolvedor ou em algum lugar na internet? Isso pode ser solucionado mediante a ajuda do Azure Artifacts. 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, dado que podemos usá-lo 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 para fluxo de CD (implantação contínua), onde a responsabilidade do fluxo e a colocação de sua aplicação no ar, em diversos ambientes, podem provocar aprovações em determinadas partes dele. Teremos um módulo mais à frente especificamente para abordar o tema.



XPe

> 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, de vez em quando, observar por que fazemos, o que fazemos e como o fazemos. Isso porque, se for mal-feito, pode afetar todo o fluxo de valor em uma organização.

Benefícios da Utilização de Controle de Versão

Primeiramente, 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 suas 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, o código-fonte é um repositório de conhecimento e compreensão inestimáveis sobre o domínio do negócio e problemas antigos resolvidos em suas aplicações, no qual desenvolvedores coletaram e refinaram através de um esforço cuidadoso. O controle de versão protege o código-fonte de catástrofe e de 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 integração contínua” (SOUZA, 2021).

Abaixo, há uma pequena lista de benefícios de se 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. Isto é, impede-os de utilizarem versões antigas, para que não haja volta a 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 na medida em que sua equipe salva novas versões do seu código.
- Reutilização: por que fazer a mesma coisa duas vezes? A reutilização de código é uma prática comum e torna a construção cada vez mais rápida e simples.
- Rastreabilidade: as auditorias não são apenas para diversão, pois, 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 diz respeito aos Sistemas da Urna Eletrônica, nos quais, em 2006, trabalhei em uma versão, e até hoje há todo um 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

Primeiramente, devo-lhes contar um pouco sobre o que é cada uma dessas tecnologias. Quando falamos Git, é importante frisar que ele foi criado em 2005, por Linus Torvalds, com intuito de ser um controle de versão inteligente e distribuído. Nele, é possível gerenciar localmente suas alterações e sincronizar com servidor a versão desejada. A maior vantagem do uso do Git é trabalhar em equipe, porém muitos desenvolvedores o utilizam para projetos individuais devido à quantidade de recursos que ele oferece.

Agora que todos já conhecem o Git, vamos contar um pouco sobre o site chamado GitHub. O GitHub é uma rede social composta por desenvolvedores. A primeira parte do nome, “Git”, deve-se à utilização do

sistema de controle de versão; já o nome “Hub” dá-se pelo conceito de integração e interconexão de pessoas do mundo todo.

Abaixo, há uma lista de comandos muito utilizados no Git, no 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, deve ser incluída com *git add*. Uma forma comum de usar `git add` é incluindo todos os arquivos que foram alterados, conforme o 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 no repositório local.
- git pull

Demonstração: instalando e utilizando Git localmente

Assista à retrospectiva da aula para acompanhar a demonstração.

Demonstração: trabalhando com Git remoto

Assista à retrospectiva da aula para acompanhar a demonstração.

Como usar Git no universo empresarial

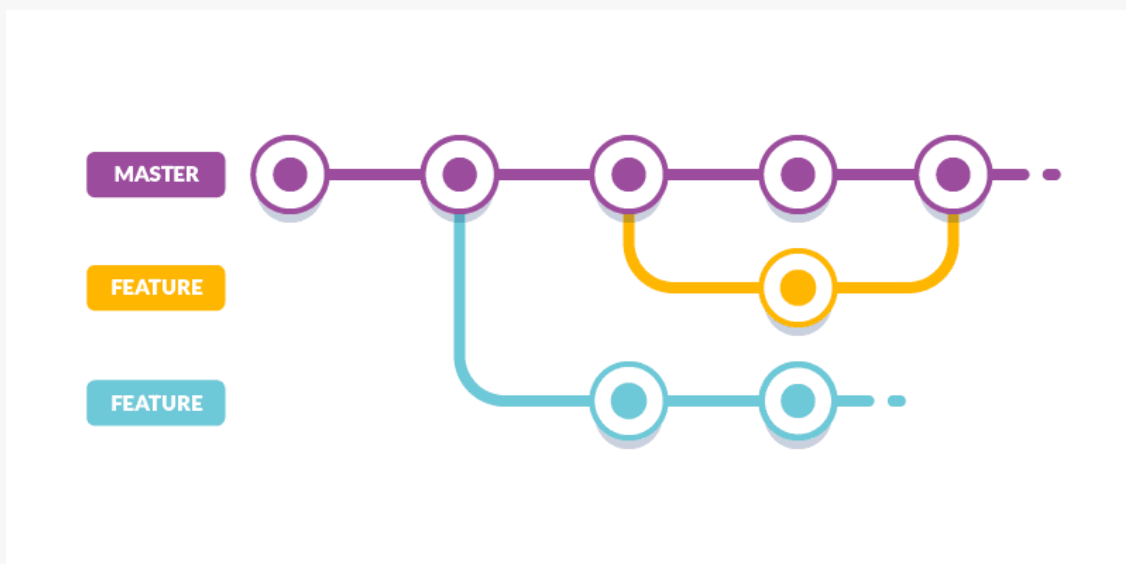
Como um sistema de controle de versão, o Git é fácil de se começar, mas difícil de se dominar. Enquanto não houver alguém para implementar o Git da maneira certa, existem várias técnicas que podem ajudá-lo a dimensionar a implementação do Git em toda a organização. Coisas simples, como estruturar seu 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. Há 2 fluxos muito comuns para implementações em ambientes corporativos. Primeiro, abordaremos o Feature branch e, em seguida, o GitFlow.

A ideia central por trás do Fluxo de Trabalho por Feature branch é que todo desenvolvimento de novos recursos ou correções deve ocorrer em um branch, mas antes precisamos explicar o que é isto. Para repositórios de controle de versão, o termo *branch* é uma ramificação do seu código, em que o código original é mantido e é feita uma cópia em uma ramificação chamada de branch. Esse encapsulamento traz mais facilidade para vários desenvolvedores trabalharem seus recursos específicos, sem que se altere 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 dada a outros desenvolvedores no que tange à aprovação de um recurso antes de ser integrado ao projeto oficial. Ou, 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. Vale dizer ainda que as Feature Branch podem (e devem) ser enviadas para o repositório central. Isso torna possível o compartilhamento de um recurso com outros desenvolvedores sem que algum código oficial seja tocado. Nesse sentido, uma vez que a Branch Master é única e especial, armazenar várias feature branches no repositório central não apresenta problemas.

Figura 2 – Feature Branch WorkFlow

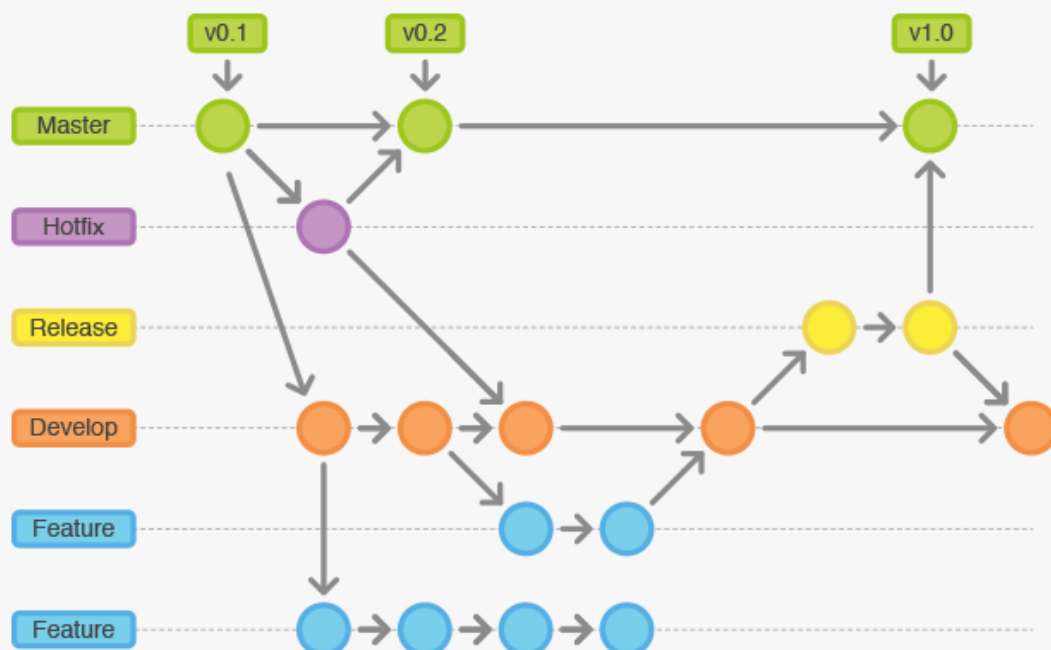


O Gitflow é um projeto de fluxo de trabalho Git, que foi publicado e popularizado por Vincent Driessen, da nvie. O fluxo de trabalho Gitflow

define um modelo de ramificação estrito projetado em torno do lançamento do projeto, de modo a fornecer uma estrutura robusta para gerenciar projetos maiores. Além disso, o Gitflow é ideal para projetos que têm um ciclo de lançamento agendado. Vale dizer ainda que esse fluxo de trabalho não adiciona novos conceitos ou comandos além do que é necessário para o fluxo de trabalho de ramo de recurso. Em vez disso, ele atribui papéis muito específicos a 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 deixamos um fluxo de como funciona. Uma vez que o pedido de pull request é feito ao repositório original ou a uma branch do repositório original, ele fará uma ação de puxar (pull) as atualizações do repositório principal master ou de um branch do próprio repositório. Abaixo, há uma ilustração de um pull request em Azure Repos.

Figura 4 – Exemplo de Pull Request

The screenshot displays the 'New Pull Request' form in Azure Repos. At the top, the repository is identified as 'FabrikamFiber / Fabrika...'. The 'Pull Requests' tab is selected. The form shows the source branch 'users/jamal/date-fix' from the 'FabrikamFiber.jamal.fork' repository being pulled into the 'master' branch of the 'FabrikamFiber' repository. The title field contains 'Added a new option to the settings page'. The description field contains 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.



XPe

> 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. Isto é, 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 em que seja possível repetir, de forma confiável e incremental, para a obtenção de software do conceito do 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 o 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 de forma contínua e a aprender onde estão situados os obstáculos para, em seguida, 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 o teste do código sempre que um membro da equipe confirma as alterações no controle de versão. Nesse sentido, a CI incentiva os desenvolvedores a compartilharem 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, devo-lhes contar uma pequena história. Há alguns anos, a TI era um departamento facilitador e que estava lá para oferecer suporte aos usuários de negócios. Isso porque o tempo provou que o software desenvolvido tinha má qualidade por padrão e que 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. Ainda 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. De um lado, departamentos de desenvolvimento são motivados a entregar valor o

mais rápido possível para manter seus clientes satisfeitos. De outro, a TI está motivada para que não mude 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 de 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, porque 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. Digo isso, porque, com a Internet, sempre temos competição global. Como se não bastasse haver concorrentes o tempo todo em nosso redor, também temos concorrentes que oferecem a melhor ferramenta para um aspecto do software que construímos. Nós precisamos entregar rápido, e o produto que fazemos deve ser bom. Também devemos fazer isso com nossa produção de software sendo barata e com alta qualidade. Para conseguirmos isso, precisamos de algo como entrega contínua (CD).

Demonstração: implementando Integração Contínua (IC)

Assista à retrospectiva da aula para acompanhar a demonstração.

Demonstração: implementando Entrega Contínua (CD)

Assista à retrospectiva da aula para acompanhar a demonstração.



XPe

> Capítulo 4



Capítulo 4. Azure DevTest Labs

O que é Azure DevTest Labs

Azure DevTest Labs permite que, em equipes, os desenvolvedores 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 para que você use e crie ambientes. Assim, 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.

Demonstração: criando seu primeiro ambiente

Assista à retrospectiva da aula para acompanhar a demonstração.



XPe

> Capítulo 5



Capítulo 5. 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 recursos relacionados à interface, que ainda estão em testes no portal.

Azure Mobile App

É uma aplicação móvel, disponível para Android e iOS, na qual podemos monitorar as assinaturas e recursos existentes em nossa infraestrutura, assim como acessar o Azure Cloud Shell, um terminal online 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 e o início do uso, 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 (MICROSOFT, 2021).

Azure CLI

A CLI do Azure (interface de linha de comando do Azure) é um conjunto de comandos usados 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 (MICROSOFT, 2021).

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. Na medida em 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 executar o código e implantar ambientes semelhantes em minutos.



XPe

> Capítulo 6



Capítulo 6. Soluções para Infraestrutura como Código (IaC) no Azure

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 toda 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.

O Ansible 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 interrelacionam, ao invés de gerenciar apenas um sistema de cada vez.

Demonstração: como implementar uma Infra as com Ansible

Assista à retrospectiva da aula para acompanhar a demonstração.

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 de 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 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.

Demonstração: como implementar um Infra as com Terraform

Assista à retrospectiva da aula para acompanhar a demonstração.



XPe

> Capítulo 7



Capítulo 7. 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 só adiciona o poder do Microsoft Azure ao seu aplicativo, mas também segurança, carga balanceamento, escalonamento automático e gerenciamento automatizado. Além disso, você 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 níveis de camadas de preço:

- Camada de testes: nesta camada de preço, os recursos são limitados. Hospeda apenas 10 aplicações e possui 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 pelo valor de R\$0,089/hora, e o número de aplicações é ilimitado.

Possui 10GB de espaço em disco, podendo realizar estabilidade manual de até 3 instâncias e, até mesmo, trabalhar com domínios personalizados.

- Camada Padrão: nesta camada de preço, os recursos são cobrados pelo valor de R\$0,466/hora, e o número de aplicações é ilimitado. Possui 50GB de espaço em disco e realiza estabilidade automática e manual de até 10 instâncias. Trabalha com domínios personalizados e habilita opção de acesso a recursos em diferentes redes.
- Camada Premium: nesta camada de preço, os recursos são cobrados pelo valor de R\$0,544/hora, e o número de aplicações é ilimitado. Possui 250GB de espaço em disco e realiza estabilidade automática e manual de até 30 instâncias. Trabalha com domínios personalizados, habilita opção de acesso a recursos em diferentes redes e possibilita integração com on-premisses.
- Camada Isolada: nesta camada de preço, os recursos são cobrados pelo valor de R\$0,466/hora, e o número de aplicações é ilimitado. Possui 1TB de espaço em disco e realiza estabilidade automática e manual de até 100 instâncias. Trabalha com domínios personalizados, habilita opção de acesso a recursos em diferentes redes e possibilita integração com on-premisses. Vale dizer ainda que o tipo de computação é isolada.

Demonstração: configurando e publicando minha primeira Aplicação Azure App Service

Assista à retrospectiva da aula para acompanhar a demonstração.

Demonstração: configurando Integração Contínua

Assista à retrospectiva da aula para acompanhar a demonstração.



XPe

> Capítulo 8



Capítulo 8. Azure Functions

O que é Serverless e como funciona

Antes de falarmos sobre Azure Functions, é necessária a explicação sobre Arquitetura Serverless. Como o próprio nome diz, é uma arquitetura sem servidor e orientada a eventos. Sua principal proposta é permitir que as empresas e times de software criem e mantenham suas aplicações serverless sem precisar se preocupar com a infraestrutura em que estão hospedadas.

Compreendendo os modelos de bind das Azure Functions

Trataremos agora de uma Introdução ao Azure Functions, que permite desenvolver aplicativos sem servidor no Microsoft Azure. Nesse sentido, 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?

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

O Azure Functions oferece suporte a gatilhos, gatilhos esses que são as maneiras 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 integra 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. A seguir, estão os serviços de integrações suportados 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)

Demonstração: implementando uma Azure Function via Portal

Assista à retrospectiva da aula para acompanhar a demonstração.

Demonstração: implementando uma Azure Function via Visual Studio

Assista à retrospectiva da aula para acompanhar a demonstração.

Demonstração: implementando uma Azure Function via Visual Studio Code

Assista à retrospectiva da aula para acompanhar a demonstração.



XPe

> Capítulo 9



Capítulo 9. Azure Logic Apps

O que são Logic Apps

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

Por exemplo, a seguir, estão apenas alguns cenários nos quais você pode realizar automação 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 for necessário estimular um sistema com alguma informação relevante.

Para construir soluções de integração empresarial com Logic Apps, você pode realizar sua escolha em uma galeria crescente, com centenas de conectores prontos para uso, que incluem serviços, como Azure Service Bus, Azure Functions, Azure Storage, SQL Server, Office 365, Dynamics, Salesforce, BizTalk, SAP, Oracle DB, ServiceNow, 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 é disparado quando um evento específico acontece, ou quando 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 agendamentos 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, 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 detectar um evento que corresponda a esses critérios, o gatilho dispara e executa as ações do workflow.

Demonstração: criando Logic Apps para receber Tweets

Assista à retrospectiva da aula para acompanhar a demonstração.

Demonstração: criando uma API POST com Logic Apps

Assista à retrospectiva da aula para acompanhar a demonstração.

Demonstração: criando uma API GET com Logic Apps

Assista à retrospectiva da aula para acompanhar a demonstração.



XPe

> Capítulo 10



Capítulo 10. Azure IoT Hub

Conceitos básicos sobre o 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 ainda mais são adicionados a cada ano.

A definição mais clara para IoT é toda e qualquer coisa a que se pode incorporar sensores e software, e, também, que se pode conectar pela Internet.

A seguir, estão as tecnologias que 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; e 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. Nesse sentido, você pode conectar milhões de dispositivos e suas soluções de back-end de maneira confiável e segura. Vale dizer ainda que quase qualquer dispositivo pode ser conectado a um Hub IoT.

Ademais, 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, suas conexões e suas falhas.

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

Além do mais, destaca-se a possibilidade de integrar o Hub IoT a outros serviços do Azure para a criação de soluções completas de ponta a ponta. Por exemplo:

- Event Grid, para que sua empresa reaja rapidamente a eventos críticos de maneira confiável, escalonável e segura.
- 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 como um todo.

Demonstração: trabalhando com emuladores IoT e IoT Hub

Assista à retrospectiva da aula para acompanhar a demonstração.

Demonstração: registrando devices IoT e enviando mensagens via C#

Assista à retrospectiva da aula para acompanhar a demonstração.

Código C# está em: <https://github.com/hsouzaeduardo/IGTI-DEMO10.2>.



> Capítulo 11



Capítulo 11. IoT Central

Introdução a 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 de gerenciamento, mantendo soluções de IoT a nível empresarial. Dessa forma, optar por construir com o IoT Central dá a você a oportunidade 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 de acordo com os insights do dispositivo, estendendo a IoT 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, além de como implementar esta alternativa de fácil gerenciamento, ao máximo, em opção de solução PaaS. É essencial, também, o conhecimento de que os recursos de IoT Plug and Play, no Azure IoT Central, encontram-se, 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.

Demonstração: como criar seu primeiro IoT Central

Assista à retrospectiva da aula para acompanhar a demonstração.



XPe

> Capítulo 12



Capítulo 12. Azure Sphere

O que é um Azure Sphere

O Azure Sphere é uma plataforma de aplicativo segura e de alto nível, com recursos internos de comunicação e segurança para dispositivos conectados à Internet. Ele é composto por uma MCU (unidade microcontroladora) cruzada, segura e conectada, por um SO (sistema operacional) personalizado de alto nível baseado em Linux (SO) e por um serviço de segurança baseado em nuvem, que fornece segurança contínua e renovável.

A MCU do Azure Sphere integra recursos de processamento em tempo real, com a capacidade de executar um sistema operacional de alto nível. Uma MCU do Azure Sphere, em conjunto com seu sistema operacional e sua plataforma de aplicativos, permite criar dispositivos seguros e conectados à Internet, que podem ser atualizados, controlados, monitorados e mantidos remotamente. Com um dispositivo conectado que inclui um MCU do Azure Sphere, seja junto ou no lugar de um MCU existente, os fabricantes de produtos obtêm oportunidade, produtividade e segurança aprimoradas. Por exemplo:

- Um ambiente de aplicativo protegido, conexões autenticadas e uso opcional de periféricos minimiza os riscos de segurança devido à falsificação, software não autorizado ou ataques de negação de serviço, entre outros.
- Atualizações de software podem ser implantadas automaticamente da nuvem para qualquer dispositivo conectado para consertar problemas, fornecer nova funcionalidade ou combater métodos emergentes de ataque, melhorando a produtividade da equipe de suporte.

- Dados de uso do produto podem ser relatados para a nuvem por uma conexão segura para ajudar a diagnosticar problemas e criar produtos, aumentando a oportunidade para serviço de produtos, interações positivas com os clientes e desenvolvimento futuro.

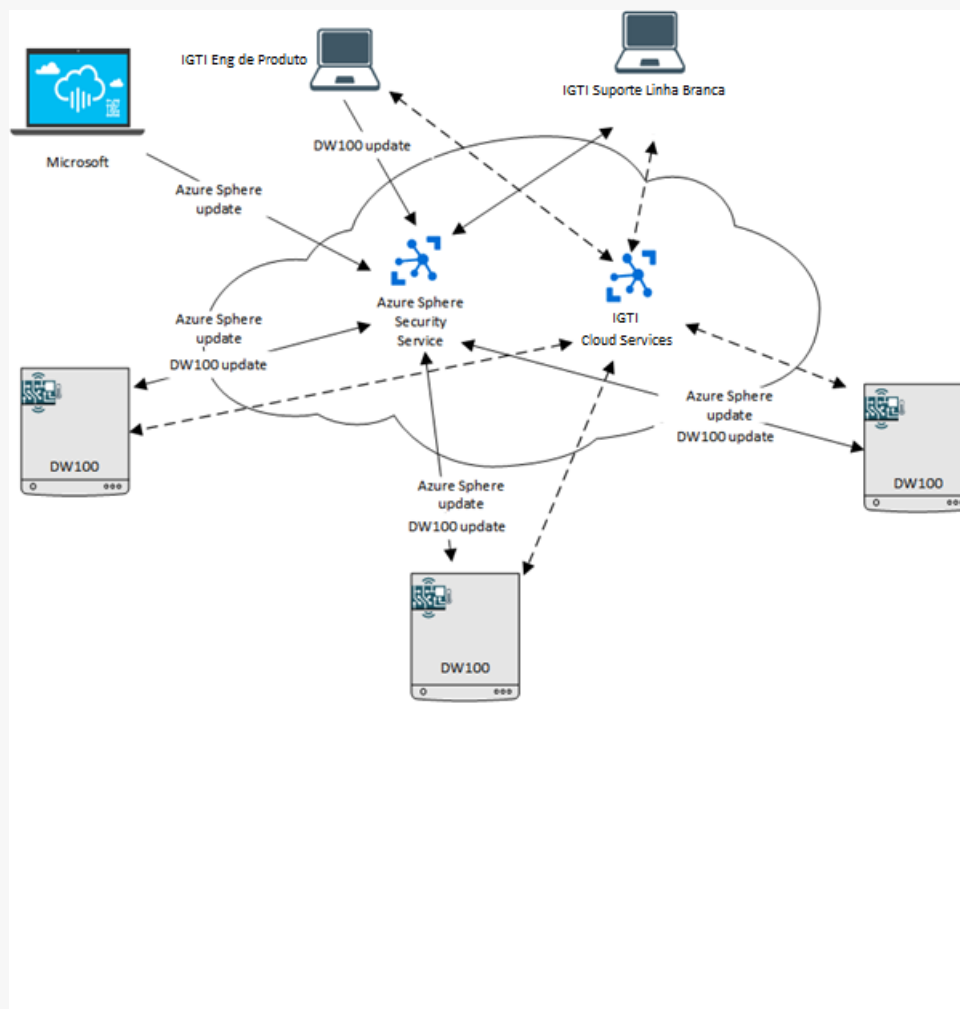
O serviço de segurança do Azure Sphere é um aspecto integrante do Azure Sphere. Usando esse serviço, as MCUs do Azure Sphere conectam-se com segurança à nuvem e à Web. O serviço garante que o dispositivo seja inicializado apenas com uma versão autorizada do software original e aprovado. Além disso, ele fornece um canal protegido por meio do qual a Microsoft pode baixar e instalar automaticamente atualizações do sistema operacional para dispositivos implantados no campo para atenuar problemas de segurança. Não é necessária intervenção do fabricante nem do usuário final, fechando uma lacuna de segurança comum.

Cenários para o uso de Azure Sphere

Para entender como o Azure Sphere funciona em uma configuração do mundo real, considere este cenário:

- O IGTI Parts Ltda. é um fabricante de linha branca que incorpora um MCU do Azure Sphere em suas lavadoras de louça. A lavadora de louça DW100 associa a MCU a vários sensores e a um aplicativo integrado de alto nível, que é executado no MCU do Azure Sphere. O aplicativo se comunica com o serviço de segurança do Azure Sphere e com os serviços de nuvem da IGTI Parts Ltda. O diagrama a seguir ilustra este cenário:

Figura 4 – Arquitetura IGTI Parts LTDA



Principais funcionalidades

Conforme Microsoft (2021), as principais funcionalidades são:

- A defesa em profundidade oferece várias camadas de proteção para ajudar a proteger os dispositivos contra ameaças e a reagir oportunamente a elas.
- A flexibilidade de implantação ajuda a manter os equipamentos existentes em segurança e a criar proteção para os novos investimentos de IoT.

- As atualizações OTA (sem fio) facilitam a adição de novos recursos e melhoram o desempenho em todos os ciclos de vida dos dispositivos.
- Os recursos de relatórios de erros e atualizações automáticas de segurança ajudam a manter você sempre à frente das mais recentes ameaças.



> Capítulo 13



Capítulo 13. 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 que automatizam responsabilidades que parecem ser mais um fardo do que a tarefa do trabalho. A Power Platform viabiliza o seu negócio para criar soluções e, simultaneamente, capacitá-lo a unir tecnologia personalizada para ajudar a todos, desde o CEO até 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, serviço de dados escalável e plataforma de aplicativos, para permitir integração e interação simples com os dados existentes. Além disso, 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. À vista disso, 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 por meio. 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 e tem 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 faça 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, sem uma interface código e sem a necessidade de cientistas de dados ou desenvolvedores.

O Power Virtual Agents aborda muitos dos principais problemas relacionados à construção de chatbots, o que faz com que seja possível eliminar a lacuna entre especialistas no assunto e as equipes de

desenvolvimento responsáveis por criá-los. Ademais, remove a complexidade de expor as equipes IA conversacional e a necessidade de escrever códigos complexos.

Nota-se, portanto, como isso 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.

Referências

MICROSOFT. Azure. [S. /], 2021. Disponível em: azure.microsoft.com/. Acesso em: 14 nov. 2022.

SOUZA, Henrique Eduardo. DevOps em uma lição de vida. [S. l.: s. n.], 2021. No prelo.