

Curso Azure DevOps LinkedIn

Sumário

1. *Da Ideia à Entrega com Azure DevOps*
 2. *Gerenciamento de Código*
 3. *Como Compilar seu Código*
 4. *Como Gerenciar seus Testes*
 5. *Azure DevOps Extensões e Integrações*
-

1. Da Ideia à Entrega com Azure DevOps

Para executar, rastrear e acompanhar os trabalhos devemos usar Work Item. Pois ele é utilizado no Azure DevOps para rastrear diferentes tipos de trabalhos.

Existem vários tipos de Work Item que podem ser rastreados, como:

- Tarefas
- Épicos
- Features
- Histórias de Usuários

Onde podemos aplicar em nossos backlogs ou aplicar em sprints posteriormente, com objetivo de trabalhar com gerenciamento de tarefas do meu time de forma mais assertiva.

O QUE SÃO WORK ITENS E SUA HIERARQUIA

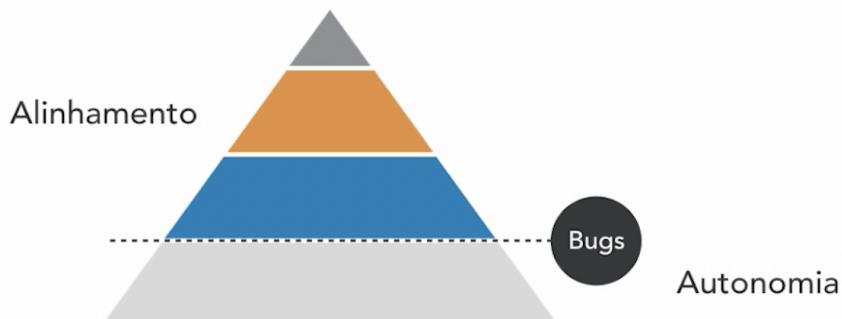
Os *Épicos* são o nível mais alto dos tipos de work itens e podem abranger vários tipos de liberações de entrega.

Abaixo temos as *Features*, que são utilizadas para determinar objetivos para um único sprint.

As *Histórias* dividem os recursos em peças de trabalho independente.

As *Tarefas* fornecem informações adicionais para histórias e são usadas para o planejamento da capacidade.

Os erros são tratados como Histórias ou Tarefas do usuário e representam problemas específicos no produto.



Épico = Sistema

Feature = Partes Específicas do Sistema

Histórias = Peças específicas de trabalho relacionada a essa feature que vamos trabalhar

Tarefas = Onde o time de desenvolvimento trabalhará

BOARDS, BACKLOGS E SPRINTS

Desafios aos planos dos projetos:

- Surpresas
- Mudança de contexto
- Objetivos não compartilhados
- Silos de Conhecimento

🔍 Backlog

É a lista priorizada de recursos que contém descrições breves da funcionalidade desejada do produto.

↔️ Sprint

É o período de tempo durante o qual um trabalho específico deve ser concluído. Onde trabalhamos os itens priorizados do backlog.

No Azure DevOps, isso é chamado de Iteração.

Primeira Sprint

Imagine que em uma primeira Sprint, nós separamos os itens do backlog e assim determinamos tudo que acreditamos que o time tem capacidade de entregar nesta sprint, especificamente.

Imagine que a lista abaixo é nosso backlog:

- Realidade** -----
- Doc: Update Create/Delete Team Project (Buck Hodges feedback)
 - Virtual Application Hosts: HostSyncService ("Host Fault-in") changes
 - Virtual Application Hosts: Host Routing reaction
 - ■ Virtual Application Hosts: Core Services Reaction / TFS Reaction
 - [SPIKE] Determine appropriate thresholds for users for public SUs
 - Push QDS data into Kusto.
 - ResourceUtilization UI notifications
 - Prep for collection the domain rollout to MSENG and the public scale units
 - Rename project named 'DefaultCollection'
 - Remove profile-only identities from the SPS configuration database
 - Make project creation automatically queue the project deletion job on failure
 - DataImport: Running custom hooks at the right time during import
 - Add L0 and L2 test to cover the ServiceHostHistoryJobExtension and prc_QueryService...
 - Clean up artifacts after a data import fails or completes
 - Set up Test MDS Account and Kusto Cluster for Resource Utilization Development
 - Data Import: Generate the import package
 - ResourceUtilization Email notifications

A linha de Realidade divide onde realmente estamos entregando nossos itens. Portanto nesse caso, não consegui entregar todos os itens na primeira sprint, o que pode acontecer.

Já em uma segunda sprint, podemos chegar no cenário abaixo:

- Doc: Update Create/Delete Team Project (Buck Hodges feedback)
- Virtual Application Hosts: HostSyncService ("Host Fault-in") changes
- Virtual Application Hosts: Host Routing reaction
- ■ Virtual Application Hosts: Core Services Reaction / TFS Reaction
- [SPIKE] Determine appropriate thresholds for users for public SUs
- Push QDS data into Kusto.
- ResourceUtilization UI notifications
- Prep for collection the domain rollout to MSENG and the public scale units
- Rename project named 'DefaultCollection'
- Remove profile-only identities from the SPS configuration database
- Make project creation automatically queue the project deletion job on failure
- DataImport: Running custom hooks at the right time during import
- Add L0 and L2 test to cover the ServiceHostHistoryJobExtension and prc_QueryService...
- Clean up artifacts after a data import fails or completes
- Set up Test MDS Account and Kusto Cluster for Resource Utilization Development
- Data Import: Generate the import package
- ResourceUtilization Email notifications

Realidade

Com o backlog completo ao final da sprint.

Cadênciā

É o ritmo ou timebox específico que os sprints seguem.

Para não acontecer como no caso acima, de acabar a sprint e ficar itens sem fazer no backlog, devemos sempre medir a cadênciā da nossa equipe (normalmente utilizando a média de tarefas entregues das duas ultimas sprints), e selecionar a quantidade de tarefas mais adequada.

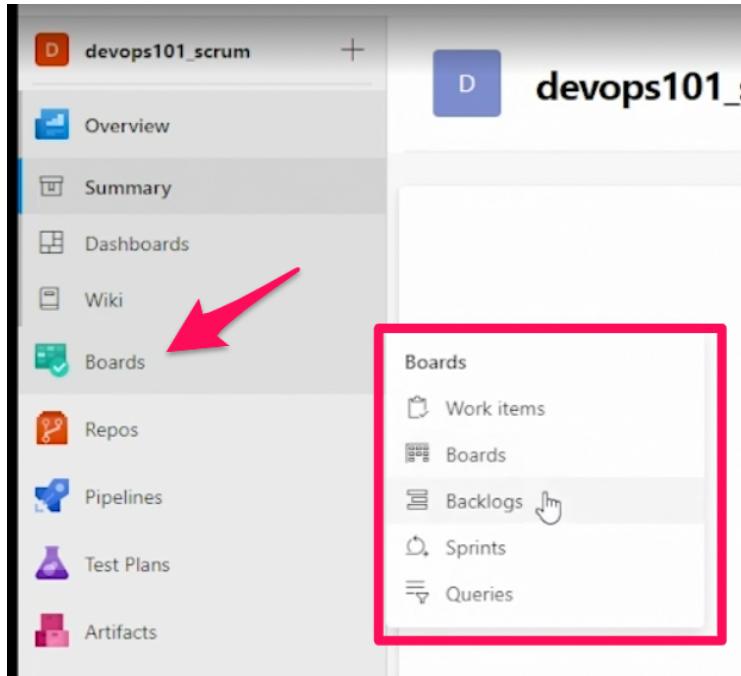
O melhor é puxar menos tarefas e caso sobre tempo, puxar mais tarefas do backlog durante o sprint, do que acabar o sprint com tarefas por fazer, pois isso pode acabar frustando a equipe.

TEAM PROJECTS, TEMPLATES E METODOLOGIAS ÁGEIS

Para utilizar um projeto com scrum, devemos ao criar o projeto, ir nas configurações avançadas da criação e ao invés de 'Basic' (como vem por padrão), devemos trocar para opção 'Scrum'.

Após realizar a criação do projeto, é necessário realizar algumas configurações para podermos trabalhar gerenciando backlogs, sprints, etc.

Para gerenciar Backlogs e Sprints vamos utilizar o 'Boards'

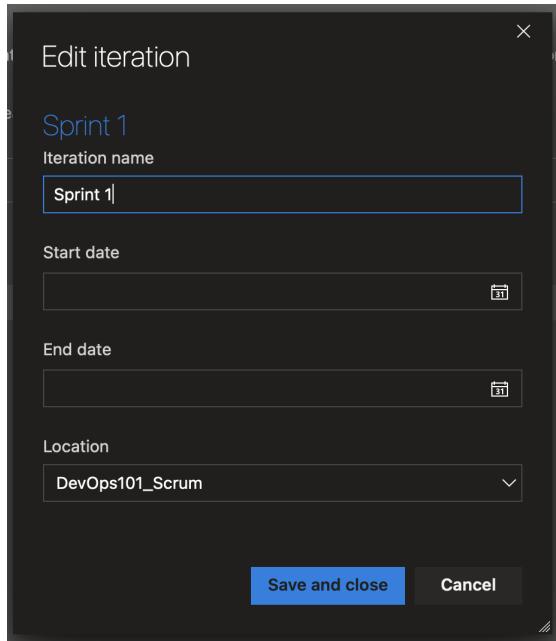


Para realizar a configuração, vamos em 'Project Settings' > 'Boards > Project Configuration'

Nesta tela vemos as sprints já criadas (como padrão) e podemos criar datas de início e fim para cada sprint.

The screenshot shows the 'Project Settings' page for the 'DevOps101_Scrum' project. The left sidebar has sections for General, Boards (selected), Project configuration (highlighted with a blue bar), Team configuration, GitHub connections, Pipelines, Agent pools, Parallel jobs, and Settings. The main area is titled 'Boards' and contains a message: 'This project is currently using the Scrum process. To customize your work item types, go to the process customization page.' Below this are tabs for Iterations and Areas. A message says: 'Create and manage the iterations for this project. These iterations will be used by teams for iteration planning (sprint planning).'. It also says: 'To select iterations for the team, go to the default team's settings.' Below this is a table with columns: New, New child, Iterations, Start Date, and End Date. The table shows iterations for the 'DevOps101_Scrum' team: Sprint 1, Sprint 2, Sprint 3, Sprint 4, Sprint 5, and Sprint 6. Red arrows point from the table to the 'Start Date' and 'End Date' headers, and to the 'Sprint 3' row.

Para adicionar data, basta clicar no botão que aparece na coluna de 'Start Date', que aparece quando você passa o mouse sobre ele, que abrirá o modal para colocar os dados.



Feito isso, teremos uma data de início e fim para essa sprint.

New	New child	+	-
Iterations	Start Date	End Date	
▼ DevOps101_Scrum			
Sprint 1	... 14/03/2022	18/03/2022	
Sprint 2			
Sprint 3			
Sprint 4			

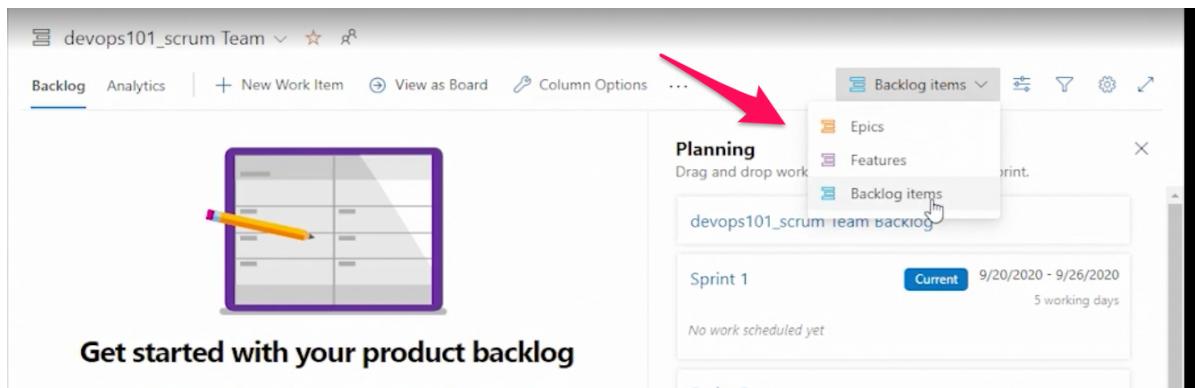
Agora podemos voltar em 'Boards', para criarmos nossas work items em nossas sprints.

PLANEJAR SEU BACKLOG

Para gerenciar seu backlogs e sprints vamos acessar nosso projeto e clicar em 'Boards'.

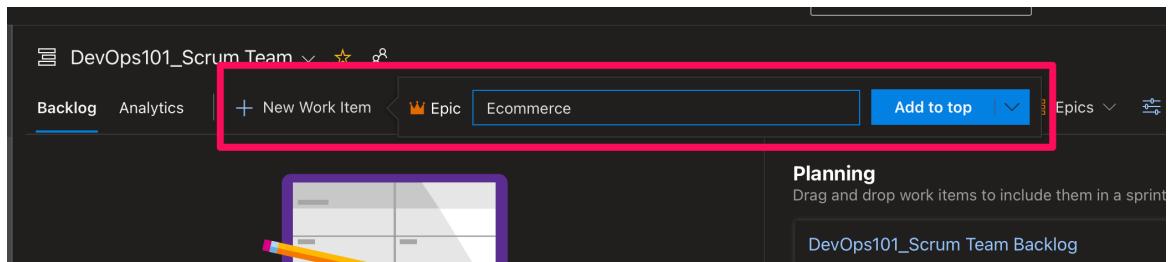
Para trabalhar com os work itens, vamos acessar dentro do boards o menu 'Backlogs', nele é onde vamos trabalhar e gerenciar todo nosso backlog. Podemos criar novas sprints também, do lado direito da tela, no menu '**Planning**'. Podemos acessar qual hierarquia queremos acessar (um ponto importante é que o épico não aparece na hierarquia, pois não vem marcado como padrão, e para

mostrá-lo devemos acessar o settings - engrenagem - e marca-lo para que ele apareça na seleção de hierarquia).



Selecionando no menu mostrado acima qual hierarquia deseja trabalhar, vamos agora criar as Work Item desejada.

Para isso vamos utilizar o botão "+ New Work Item", onde o sistema abrirá a caixa de texto para digitarmos o nome do épico, feature ou backlog e a opção de onde queremos inserí-lo (caso a lista já esteja grande).



Criamos o Épico como Ecommerce como nosso sistema principal.

Agora precisamos criar nossas Features. E para isso vamos clicar no "+" ao lado esquerdo do Épico.

The screenshot shows the Azure DevOps Backlog screen for the 'DevOps101_Scrum Team'. At the top, there are navigation links for 'Backlog' (which is underlined), 'Analytics', 'New Work Item', 'View as Board', 'Column Options', and more. Below the header, there's a table with columns for 'Order', 'Work Item Type', 'Title', and 'State'. A single row is visible, labeled '1 Epic Ecommerce State: New'. A red arrow points to the '+' button on the left side of the table.

O sistema vai abrir para preenchermos a tela de preenchimento de Features.
No caso vamos cadastrar o Checkout do sistema.

The screenshot shows the 'Create Work Item' dialog for a 'Feature' type. The title is 'FEATURE 5' and the sub-type is 'Checkout'. The work item has been created with the status 'New'. It includes fields for 'Description' (with placeholder 'Click to add Description'), 'Acceptance Criteria' (with placeholder 'Click to add Acceptance Criteria'), and 'Discussion' (with placeholder 'Add a comment. Use # to link a work item, ! to link a pull request, or @ to mention a person.'). On the right, there are sections for 'Status' (with 'Start Date' and 'Target Date' fields), 'Deployment' (with a note about tracking releases), 'Details' (with 'Priority' set to '2', 'Effort', 'Business Value', and 'Time Criticality'), 'Development' (with a note about linking to Azure Repos), and 'Related Work' (with 'Add link', 'Parent', and a link to 'Ecommerce').

Cadastrada a feature, precisamos criar agora o backlog. Para isso vamos clicar no "+" ao lado esquerdo da Feature.

The screenshot shows the Microsoft Azure DevOps interface for the 'DevOps101_Scrum Team'. The 'Backlog' tab is selected. At the top, there are navigation links for 'Backlog' (highlighted in blue), 'Analytics', 'New Work Item' (with a plus sign icon), 'View as Board', 'Column Options', and three dots for more options. Below the header is a table with columns: 'Order', 'Work Item Type', 'Title', 'State', and 'Reason'. One row is visible: '1', 'Epic', 'Ecommerce', '● New', and '● New'. Below this table is a large dark area with a red arrow pointing to a blue '+' button on the left side.

Nesse ponto podemos escolher criar:

- Product Backlog Item
- Bug

Vamos escolher Product Backlog Item (PBI).

O sistema abrirá a tela de cadastro de PBI.

The screenshot shows the 'NEW PRODUCT BACKLOG ITEM' creation form. The title bar says 'NEW PRODUCT BACKLOG ITEM *'. The main area has a text input field containing 'Implementação de nova versão x da página de checkout|'. Below it are sections for 'Description' (with placeholder 'Click to add Description'), 'Details' (Priority: 2, Effort, Business Value), 'Deployment' (instructions to track releases), 'Development' (button to add links), and 'Related Work' (listing a parent item and a task named '5 Checkout' with status 'Updated 6 minutes ago, ● New').

Porém para que o time de desenvolvimento consiga trabalhar com esse backlog, é preciso que sejam criadas as tarefas (tasks).

Para criar as tasks, vamos clicar no "+" ao lado esquerdo do PBI.

A screenshot of the Azure DevOps Backlog board for the 'DevOps101_Scrum Team'. The backlog table shows an epic 'Ecommerce' and a feature 'Checkout'. A red arrow points to the '+ New Work Item' button at the bottom left of the table. To the right, a sidebar titled 'Planning' shows 'Sprint 1' and 'Sprint 2'.

O sistema abrirá a tela de cadastro de cadastro de New Task.

A screenshot of the 'New Task' creation dialog. The task title is 'Testes'. The 'Details' section shows a priority of '2' and remaining work of '8'. A red arrow points to the 'Priority' field. The 'Deployment' section contains deployment status reporting information. The 'Related Work' section shows a linked work item.

Nele eu posso cadastrar a task, o esforço para a atividade (em horas).

A screenshot of the Azure DevOps Backlog board. The backlog table now includes three tasks under the 'Product Back...' epic: 'Testes', 'Deployment', and 'Teste em Ambiente de Produção'. The 'Testes' task is currently selected, indicated by a blue background.

Pronto, agora que já cadastrei as tasks, temos o backlog pronto para realizar a Planning.

SPRINTS EFICAZES

Para o planejamento da sprint (planning), vamos utilizar a mesma tela de Backlog. Para incluir uma task ou PBI em uma sprint, basta segurar com o clique do mouse e arrastá-la para a sprint desejada. Feito isso, a task ou PBI estará na sprint que colocou.

No menu 'Sprints', podemos ver o quadro da nossa sprint, podemos definir os membros da nossa equipe e também o capacity dela.

Caso um membro não trabalhe alguns dias/horas, podemos cadastrar esse tempo, pois assim ele não é considerado e não corremos o risco de contar com um colaborador que não estará na sprint por "x" motivo ou tempo.

Podemos também definir qual atividade aquele membro faz na sprint e qual capacidade (em horas) ele tem para atuar por dia.

Em Taskboard temos a visão de Kanban da nossa sprint, onde podemos arrastar os cards.

MÉTRICAS E QUERY

Os desafios aos planos do projeto quando se fala em DevOps, o principal deles é o:

- Continuos Deployment

Essa é uma das principais práticas que se deve trabalhar quando falamos da cultura de DevOps.

Realizar entregas contínuas do seu software de features, de um ambiente específico de forma automáticos e validações ao longo desse ciclo. Porém para atingirmos esse nível de maturidade precisamos entender como está o ciclo do nosso projeto, e o planejamento de um projeto é o ponto de partida quando falamos de trabalhar com ciclo de vida de um software.

Para entendermos o nível de maturidade da nossa equipe, podemos trabalhar com algumas métricas dentro do Azure DevOps.

Para isso podemos utilizar as Query.

- Queries são um conjunto de cláusulas que ajudam a encontrar itens de trabalho que um usuário pode revisar, triar, atualizar ou listar em um relatório.
-

BOAS PRÁTICAS COM AGILIDADE

Devemos sempre trabalhar com o framework junto com o DevOps, para melhorar a efetividade da ferramenta.

2. Gerenciamento de Código

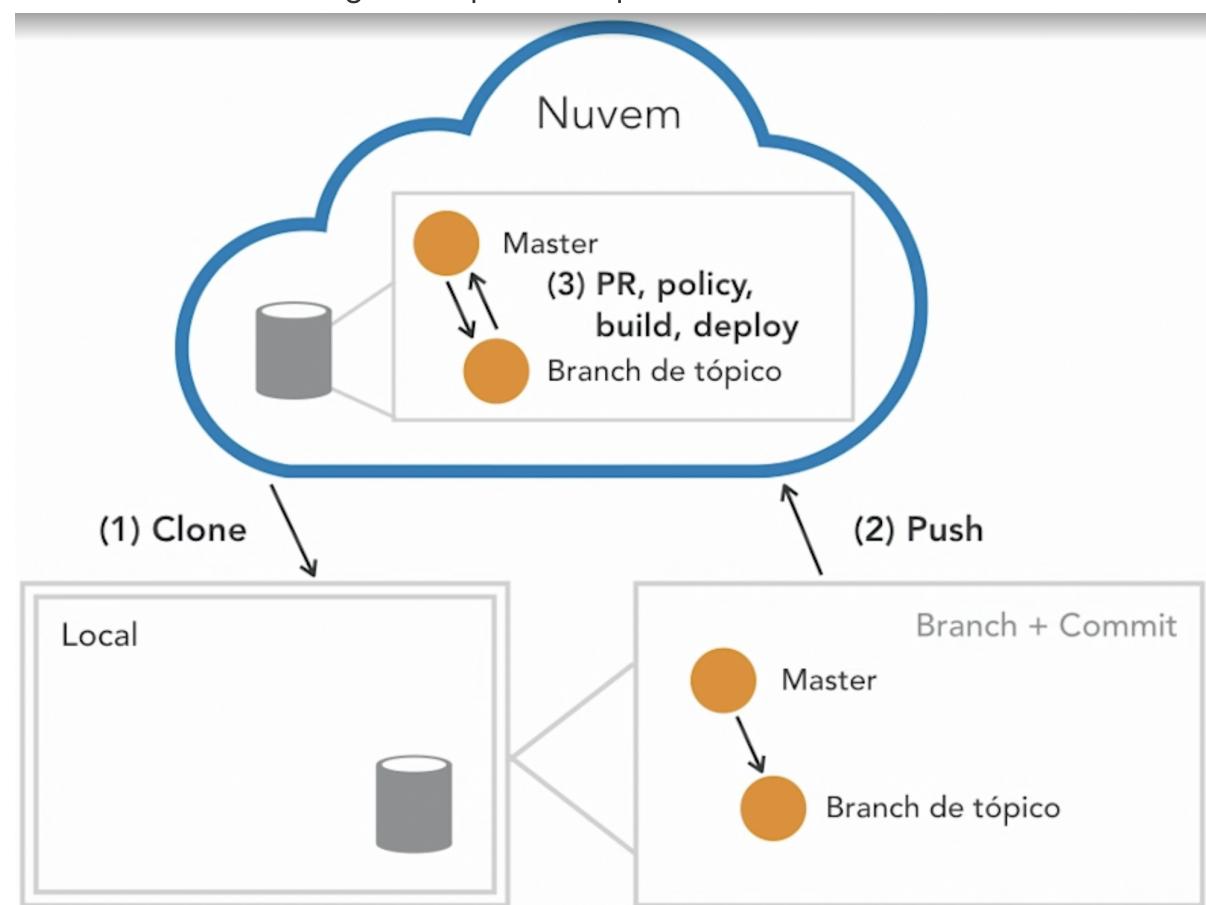
APRENDER GERENCIAMENTO DE CÓDIGO

No gerenciamento de código, podemos gerenciar milhares de arquivos, assim como várias extensões desses arquivos.

E com isso, realizar uma integração e colaboração entre times de desenvolvimento.

Para isso, utilizamos o Git.

Podemos criar um código ou copiar um repositório externo.



COMO VERSIONAR UM CÓDIGO NO AZURE REPOS

Repositório Git

Um repositório, é um diretório ou espaço de armazenamento onde vivem seus projetos.

Podemos ter um ou mais repositórios Git.

Commit

Comando para salvar o status das alterações feitas no repositório em um ponto no tempo.

Sendo assim, a cada nova alteração que fazemos, realizamos um commit.

Pull Request

Processo que pode ser usado para revisar o código em uma ramificação de recurso, antes de mesclá-lo em outra ramificação (master).

Após realizar o commit, podemos fazer o pull request. Como boa prática. Pois assim podemos fazer alguns testes específicos e a revisão do nosso código.

IMPORTE UM REPOSITÓRIO DE OUTRO GIT

Para utilizar o versionamento de código, devemos utilizar o menu do sistema chamado 'Repos'.

Nesse menu teremos nosso primeiro repositório, porém vazio.

Aqui podemos ver os repositórios já existentes, criar um novo repositório ou importar.

CLONE SEU REPOSITÓRIO DE CÓDIGO

Para usar o código localmente, precisamos acessar no azure > 'Repos'.

Vamos clonar através do Visual Studio Code.

Clicamos no botão "Clone", no canto superior direito da tela.

The screenshot shows a GitHub-style interface for a repository named 'DevOps101_Scrum'. On the left is a sidebar with repository files: .github, WebApplication, .gitignore, azure-pipelines.yml, README.md, SECURITY.md, and WebApplication.sln. The main area shows a table of files with columns for Name, Last change, and Commits. A red arrow points to the 'Clone' button in the top right corner of the main content area. Below the table, there's a section titled 'DevOps 101' with language information: 'Language: ASP.NET Core' and 'AZ-400 - Dia 2'.

Name	Last change	Commits
.github	3 de jan.	bca7e73b Add or update the Azur...
WebApplication	24 de ago. de 2021	bfd52d18 Update Index.cshtml J...
.gitignore	19 de jan. de 2019	8676ac14 Initial commit Jaquelin...
azure-pipelines.yml	1 de out. de 2021	d73de3f5 Update azure-pipeline...
README.md	23 de nov. de 2021	c5c72db5 Update README.md J...
SECURITY.md	14 de ago. de 2020	e1851c2b Create SECURITY.md J...
WebApplication.sln	28 de jul. de 2020	7a9d7225 change jaquecr

APlique seu primeiro Commit Local e Remotamente

Primeiramente precisamos criar uma Branch/Feature.

Ela pode ser criada tanto no Visual Studio Code ou no Azure DevOps.

Após realizar a criação da branch, precisamos alterar nosso repositório apontando para a branch criada, realizar a alteração desejada e fazer o commit (Push e Pull).

Pronto, se olhar para o site do Azure DevOps a nova branch com a alteração realizada, já deve constar lá.

CONCEITOS DE PULL REQUESTS

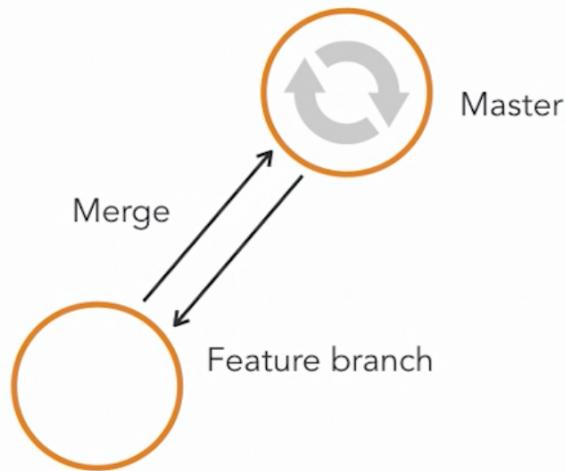
O Pull é uma maneira de aplicarmos políticas e evitarmos que um código incorreto siga para uma branch específica, seja ela master, develop, etc.

Com ele, podemos aplicar as políticas:

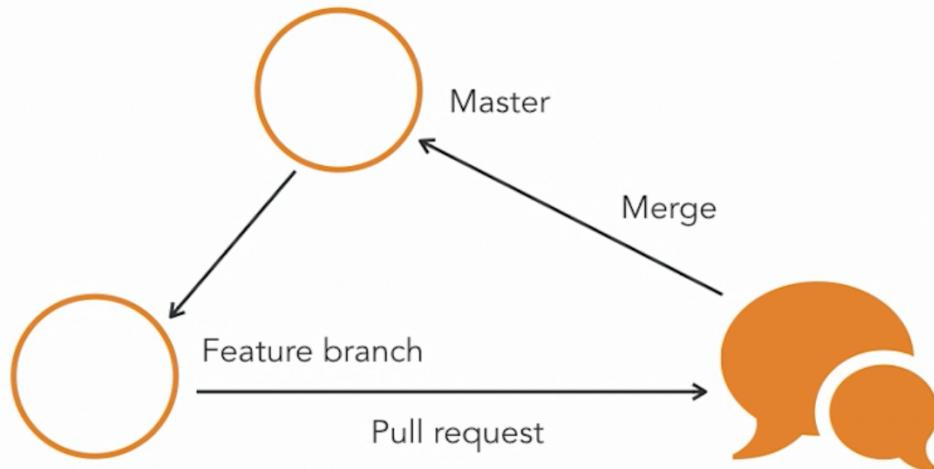
- Incluir um build para compilar e testar esse código antes que ele passe para uma branch;
- Política de Work item;

- Revisor de código.

O mais comum é quando criamos uma branch, desenvolvemos e depois voltamos isso para a master fazendo o merge. Como mostra a imagem abaixo.



Com o Pull request, também será criada a branch, porém antes de realizar o merge, o código passará pela validação de política do Pull e só depois de aprovado que o merge é realizado na master. Como a imagem abaixo.



PULL REQUEST E AZURE REPOS

Para aplicar o primeiro Pull, precisamos entender a alteração que você realizou.

No site do Azure DevOps, temos um menu na seção 'Repos' chamado 'Pull requests'.

A screenshot of the Azure DevOps interface, specifically the 'Pull requests' section for the 'DevOps101_Scrum' repository. The left sidebar shows navigation options like Overview, Boards, Repos, and Pull requests, with a red arrow pointing to the 'Pull requests' link. The main content area displays a message: 'No pull requests match the given criteria'. It includes a brief description of what pull requests are used for and a 'New pull request' button.

Nesse caso, não temos nenhum Pull request em aberto.

Aqui é onde criamos as regras de Pull, para uma branch Master (por exemplo).

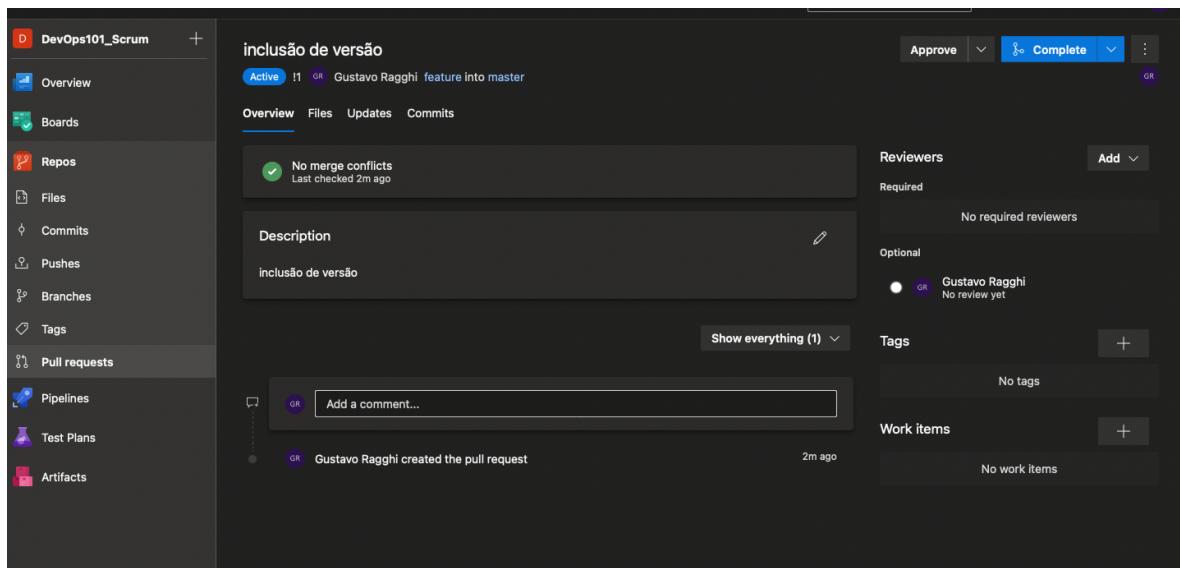
****Lembrando que branch master NUNCA deve receber código diretamente, sem uma validação****

Clicando em "New Pull request" o sistema abre a tela para você criar o Pull request puxando a ultima alteração realizada, inclusive com a descrição que foi colocada no commit.

Após preencher as informações (aprovador, work item, tags), tem opção de ver os arquivos que foram alterados e quais são os commits que estão nesse Pull request.

Após salvar, o Pull é criado.

Ele já valida se existe conflitos também.



Podemos também adicionar comentários, para tratar algum ponto que não está ok.

Antes de completar, precisamos aprovar-lo.

BOAS PRÁTICAS PARA ESTRATÉGIAS DE BRANCHES E PULL REQUESTS

- Criar *feature branches* para trabalhar em novas funcionalidades;
- Use sintaxe comum para nomes de *branches* para identificar o *owner* e o escopo;
- Use nomes descritivos para commits, víncule work item;
- Alavanque a utilização de *pull request* para controle de qualidade e dependências;
- Delete *branches* depois do merge.

3. Como Compilar seu Código

APRENDA A COMPIRAR SEU CÓDIGO

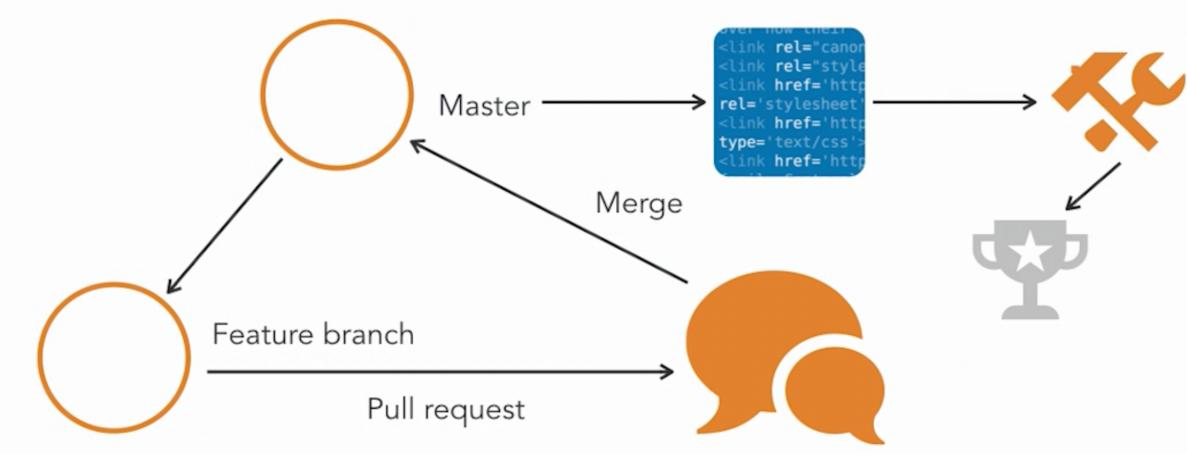
No Azure DevOps temos a opção *Continuous Integration*.

Continuous Integration é a Capacidade de criar e gerenciar processos de compilação que compilam e testam automaticamente seus aplicativos na nuvem.

Com essa prática, a cada novo commit, ele realiza a compilação, realiza os testes.

É uma ótima opção para realizar os processos de forma automática.

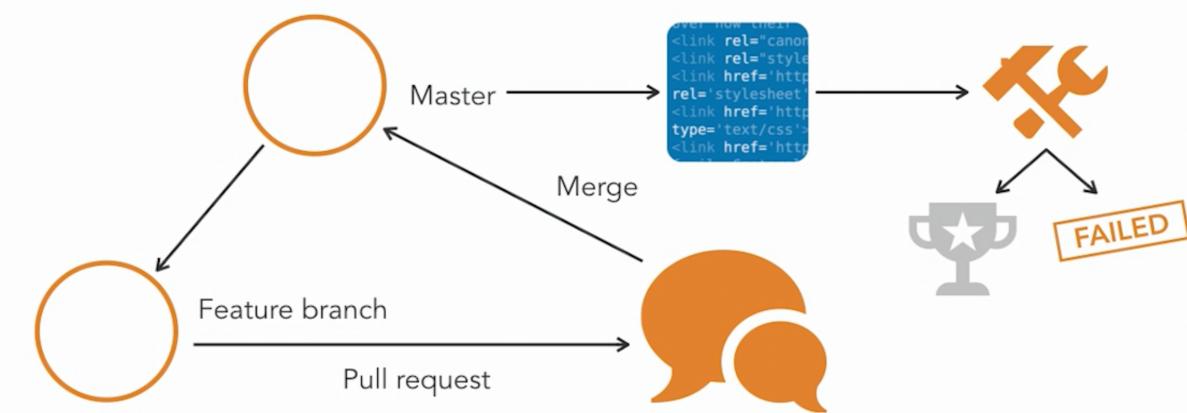
Compilando seu código



🔍 Agentes de Compilação

É um item encarregado de criar, complicar, testar e implantar seu código na nuvem.

Compilando seu código



Benefícios do *Continuous Integration*

- Confiança de que as alterações de código se integram bem a outras alterações mescladas;
- Se a compilação falhar, uma ferramenta unificada ajudará as equipes a identificar código mesclado com problemas

COMO CRIAR UM TEMPLATE DE BUILD

Quando criamos um build, vamos trabalhar com modelo de agentes. Para realizar a compilação desse código.

O agente é baseado no sistema operacional ou outros componentes.



Após a configuração inicial, vamos construir nosso pipeline.

Pipeline de Build: É o conjunto de etapas e tarefas que o processo de build deve concluir para que o código seja construído com êxito.

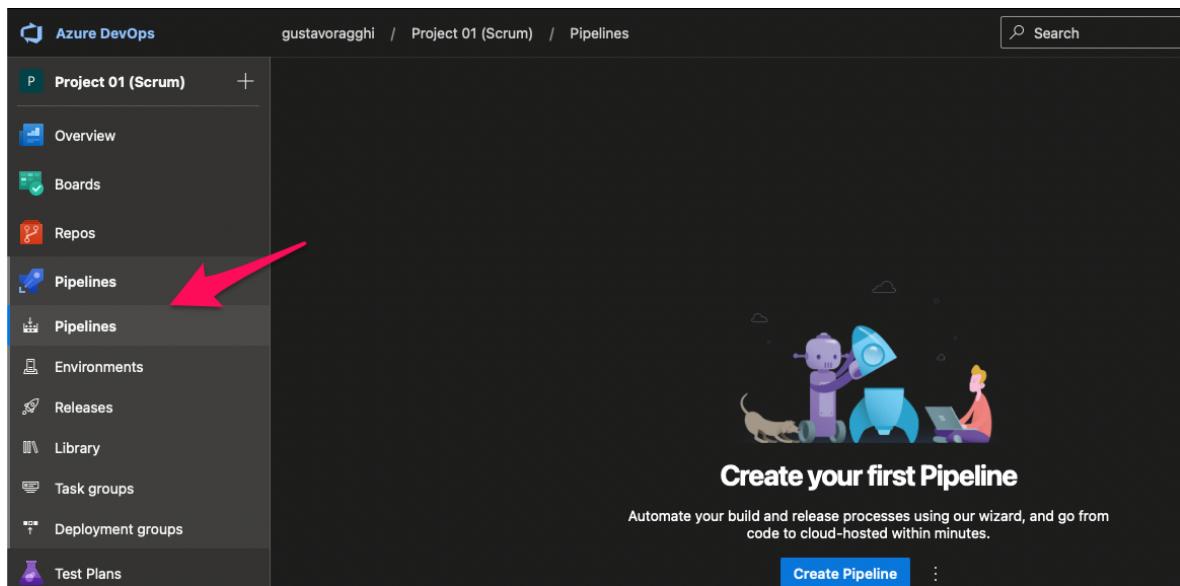
Dentro do pipeline de build podemos trabalhar com Uma ou Mais etapas.

Etapa de Build: Ação executada para concluir o processo de build.

Cada etapa ou ação que vc executa, representa: teste automatizado, compilação, deployment, etc.

PIPELINE DE BUILD E AZURE PIPELINES

Para compilar o código, precisamos criar uma Pipeline. Para isso vamos acessar o menu 'Pipelines', na lateral esquerda do Azure DevOps.



Aqui temos a opção de Criar a Pipeline.

Após clicar em "Create Pipeline", o sistema vai perguntar onde nosso código está hospedado.

Precisamos seguir os passos para a configuração do Pipeline, e nosso pipe estará configurado e pronto para ser executado.

GITHUB E AZURE PIPELINES

Podemos usar o GitHub para pipeline de realese.

Para isso precisamos criar um arquivo **YAML**.

YAML: É uma linguagem de serialização de dados legível por humanos; é comumente usada para arquivos de configuração.

Podemos declarar todos os itens de configuração como etapas, tarefas necessárias para build e até release. Seja em ambiente de desenvolvimento ou produção.

AZURE PIPELINES E SERVICE CONNECTIONS

Para trabalhar com integração de outras ferramentas externas, devemos ir em "Project Settings" > 'Service conections'.

Aqui podemos criar novas conexões com Azure, BitBucket, Docker, GitHub.

Após criar a conexão, quando criar um novo pipeline com a opção conectada o sistema já apresentará o que está disponível no repositório conectado.

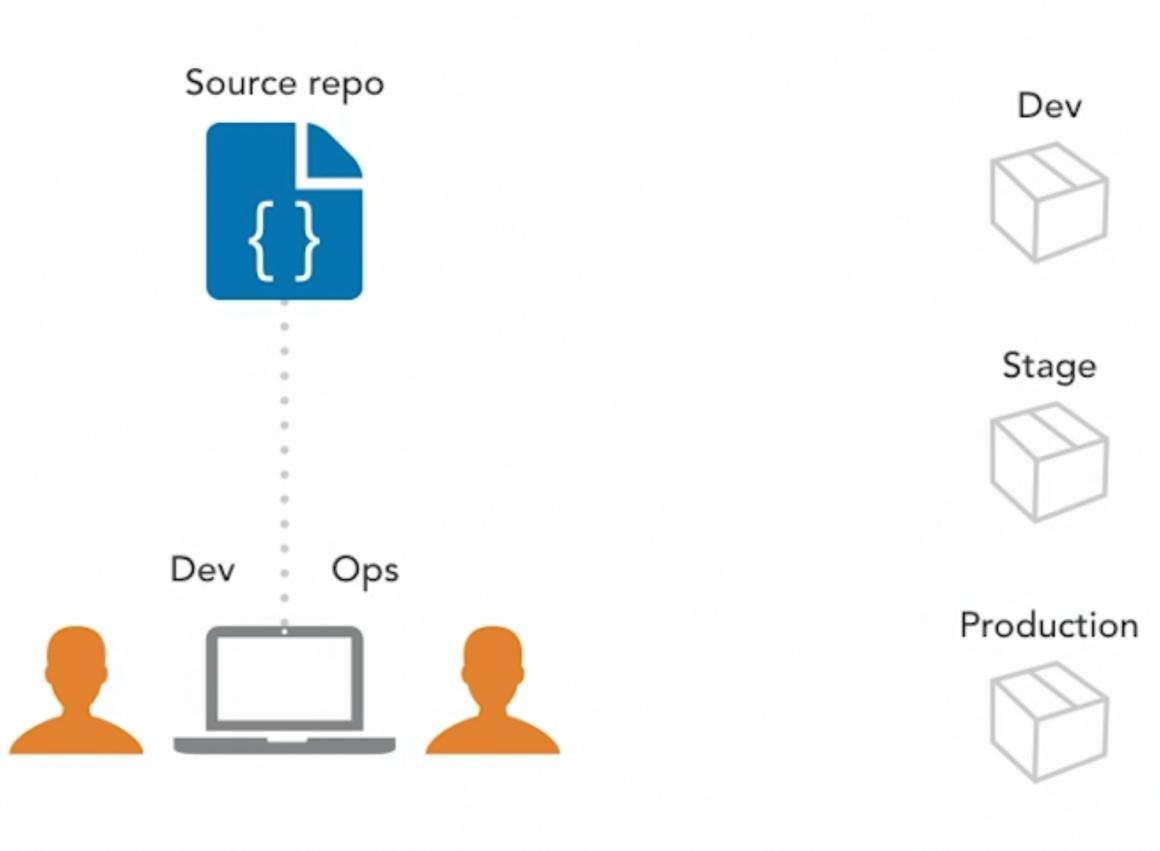
CONCEITOS DE PIPELINE DE RELEASE

Releases são os pacotes que geram posteriormente a compilação do seu código em uma build (template de build), ou também um pipeline de build.

Após a compilação, podemos entregar esse pacote em qualquer ambiente (DEV, QAS, PRD).

Assim, esse time pode realizar seu trabalho, fazer os commits e depois aplicar o versionamento de código.

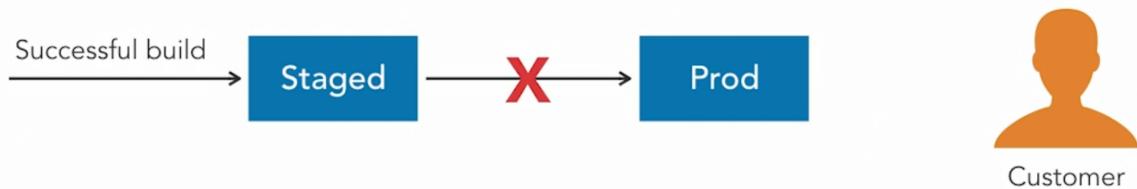
Posteriormente realizamos o pipeline de build.



Para executar esse gerenciamento de release precisamos lembrar que esse processo é utilizado para impulsionar o desenvolvimento, teste, implantação e suporte de versões de software.

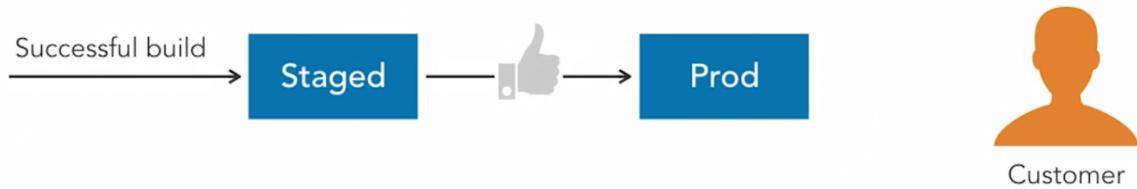
Executando esse pipeline de build, conseguimos entender como é o processo e o fluxo de uma branch de master até o pull request até voltar para a master e compila com sucesso.

Se ela falhar, o processo de release não continua.

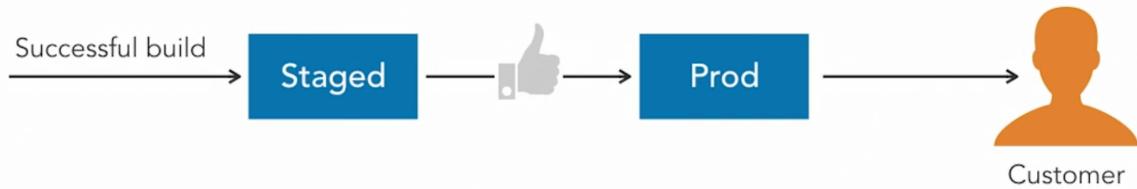


Ao falhar, como na imagem acima ele voltará para a etapa anterior aguardando a correção e um novo envio. Sendo assim, ele não chegará ao cliente.

Após realizar a correção e executá-lo novamente, com o sucesso, o pacote é aplicado no ambiente (PRD no caso da imagem).



E fica disponível para o cliente.



AZURE E AZURE PIPELINES

Vamos trabalhar com conceito de Azure Pipeline e Nuvem (Azure).

Nesse caso precisamos entender sobre Continuous Deployment. Que na maioria das vezes é que após você criar seu pacote, com todos os seus arquivos binários, arquivos

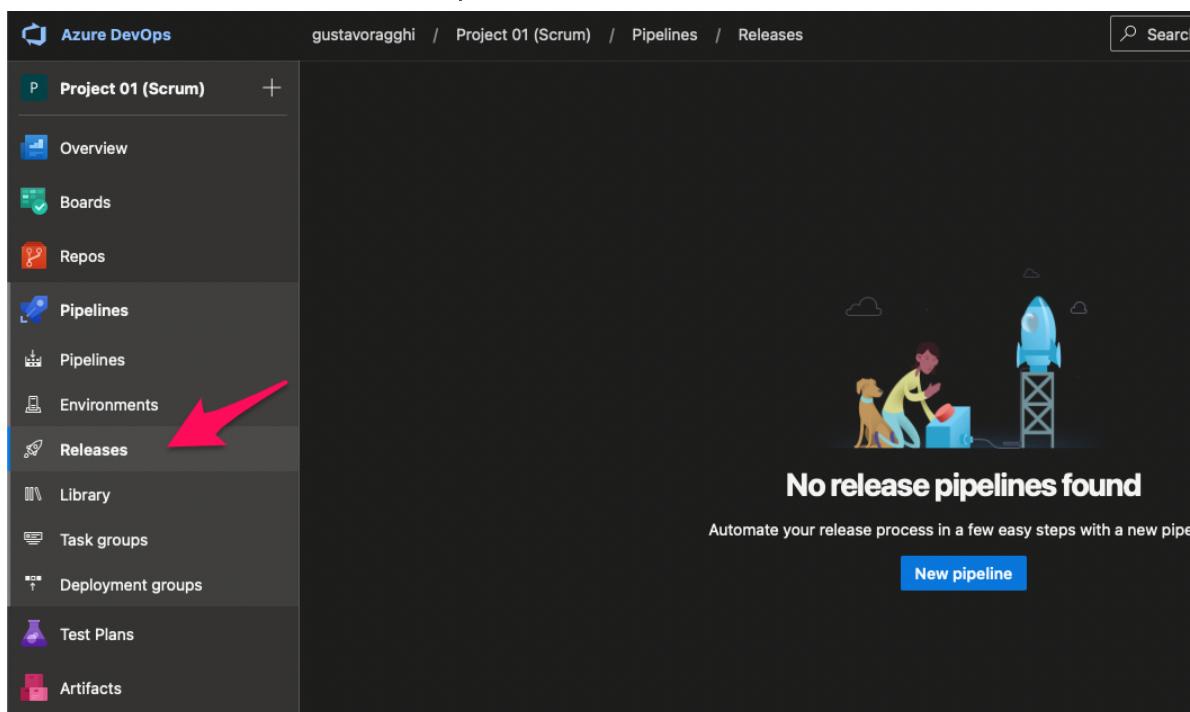
necessários para entregar em um ambiente ele o faz de forma automática e a todo momento.

Com isso estamos buscando uma maior qualidade em relação ao nosso código, testes e ao nosso software como um todo.

Release Pipeline: Uma coleção de ambientes que pode conter uma ou mais tarefas.

CRIAR UM PIPELINE DE RELEASE

Para criar uma releases, vamos acessar o menu "Pipelines" > 'Releases', no lado direito do menu do Azure DevOps.



Acessando esse menu, temos a opção de criar um novo pipeline (botão "New Pipeline"), porém de release.

Clicando em "New Pipeline" temos alguns templates padrões.

Artefato que devemos selecionar, nada mais é do que nosso pacote que vamos adicionar.

COMO ENTREGAR SEU CÓDIGO NO AZURE

Para criar no Azure, precisamos criar um Serviço de Aplicativo (APP Service) no Azure, caso ainda não possua.

4. Como Gerenciar Seus Testes

APRENDA A GERENCIAR SEUS TESTES

Para melhorar a qualidade do nosso código, o Azure possui uma feature chamada Azure Test Plans.

Nele você pode gerenciar, fazer o planejamento dos testes, executá-los de forma manual ou automatizada.

É possível também criar dashboards, para gerar relatório sobre os testes que você precisará realizar ao longo do seu projeto.

PLANEJAMENTO DE TESTE COM AZURE TEST PLANS

No planejamento de testes, podemos trabalhar com um ou mais tipos de work items.

Tipos de Workitens

Os planos de testes agrupam **Test suites** e casos de **Test cases** individuais. Os planos de teste incluem **static test suites**, **requirement-based suites** e **query-based suits**.

Dentro dos Test suites temos os Test cases. Podemos ter um ou mais grupos de test case dentro do test suit.

Testes cases são tipos de workitens que podemos trabalhar com cenários de testes, separadamente ou em um único plano de teste.

Existe possibilidade de clonar esses test case para outros planos de testes, e posteriormente executá-lo, manualmente ou automaticamente.

Os **Test cases** validam partes individuais da implantação do seu código ou aplicativo. Você pode garantir que seu código funcione corretamente, não tenha erros e atenda aos requisitos de negócios e de clientes. Você pode

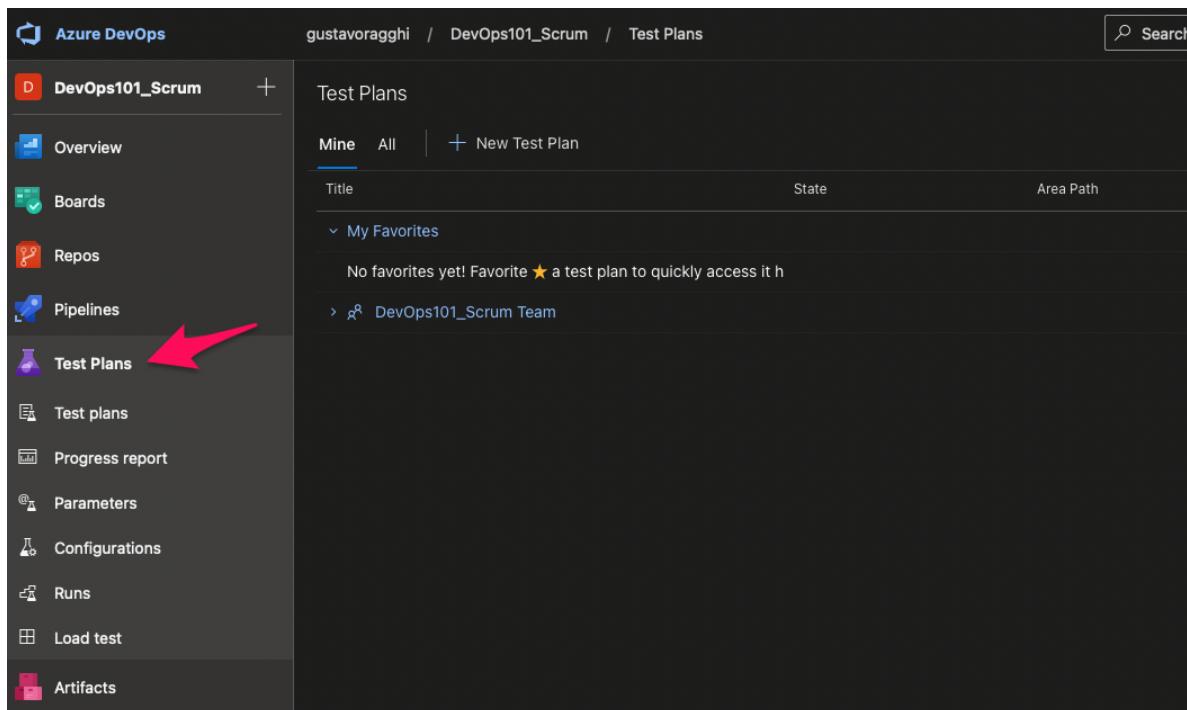
adicionar **Test cases** individuais a um plano de teste sem criar um **Test suit**, se desejar.

INICIAR COM AZURE TEST PLANS

Para iniciar, é preciso habilitar pois o Test Plans é cobrado a parte.
Você pode habilitar o teste por 30 dias.

COMO EXECUTAR TESTES MANUAIS E AUTOMATIZADOS

Após habilitar, podemos acessar o menu 'Test Plans' ao lado esquerdo da tela.



The screenshot shows the Azure DevOps interface for the 'DevOps101_Scrum' project. The left sidebar lists various options: Overview, Boards, Repos, Pipelines, Test Plans (which has a pink arrow pointing to it), Test plans, Progress report, Parameters, Configurations, Runs, Load test, and Artifacts. The main area is titled 'Test Plans' and shows a table with columns for 'Title', 'State', and 'Area Path'. A message indicates 'No favorites yet! Favorite ★ a test plan to quickly access it'.

Clicando em Test plans, você pode criar os seus Test suits e Test cases e depois realizar a execução de cada um. Sendo ele Web ou Desktop.

The screenshot shows the Azure DevOps Test Plans interface. On the left, there's a sidebar with various options like Overview, Boards, Repos, Pipelines, etc. The main area is titled 'DevOps101_2022 (ID: 11)' and shows a 'Test Suites' section with 'DevOps101_2022 (1)'. Below it, under 'Test Points (1 item)', there's a table with one row:

Title	Outcome	Order
Validação Checkout	Failed	1

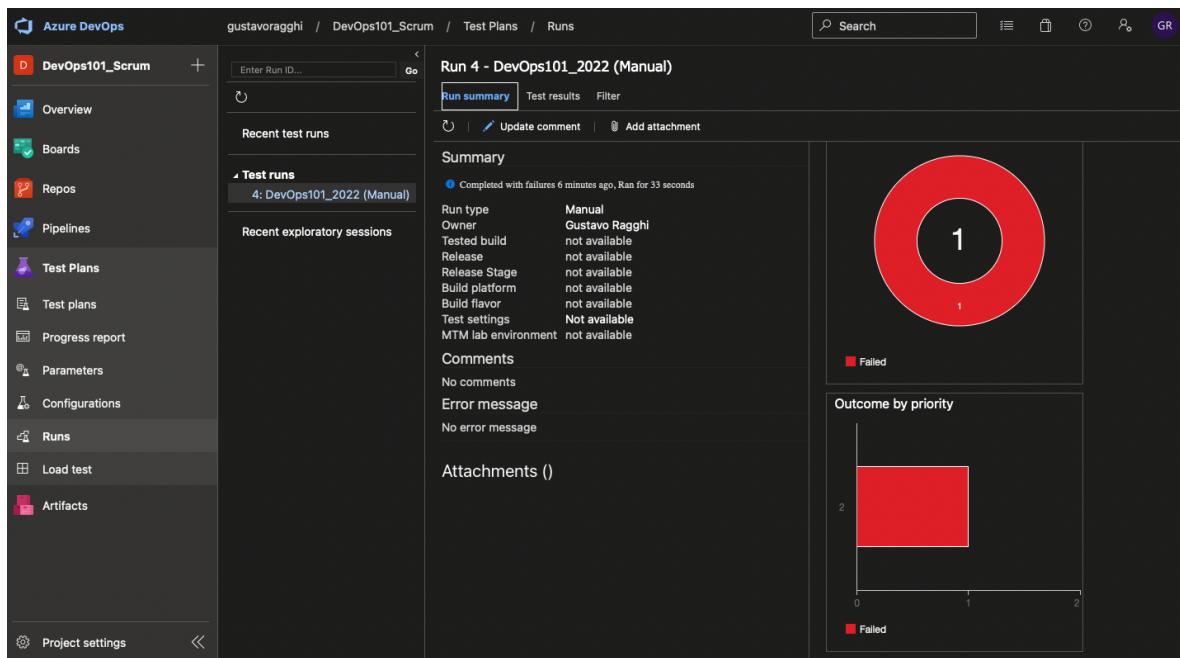
Na tela de execução, você pode ir acompanhando os passos, criando bugs, gravando a tela ou tirando print caso necessário.

The screenshot shows the '12: Validação Checkout' test execution screen. It lists the following steps:

- Entrar no site
EXPECTED RESULT: Site no ar
- Navegar pela página de produtos
EXPECTED RESULT: Página de produtos ok
- Selecionar um produto
- Ir para o carrinho
- Finalizar compra na checkout
-

Each step has a checkbox and a circled X icon next to it, indicating the current status of the test point.

Após a realização dos testes, acessando o menu 'Runs', é possível verificar os Dashboards, métricas e relatórios das execuções realizadas.



Caso necessite automatizar seu teste, após criá-lo, é necessário adicionar ao pipeline de execução.

CRIAR E CUSTOMIZAR SEU DASHBOARD

No menu 'Progress Report', conseguimos visualizar, criar e customizar dashboard de testes.

Posso criar um Dashboard de testes também no menu Overview.

AZURE DEVOPS DE PONTA-A-PONTA

Podemos trabalhar com ele desde quando planejamos o software, até quando versionamos o software, até a build, release, etc.

Devemos passar sempre para o time cada feature do Azure DevOps, deixando sempre claro, não gerando confusão entre as ferramentas que podem ser utilizadas.

5. Azure DevOps Extensões e Integrações

APRENDA AZURE DEVOPS EXTENSÕES E INTEGRAÇÕES

Acessando extensões e integrações, podemos executar alguns builds e releases de formas diferentes.

Extensão

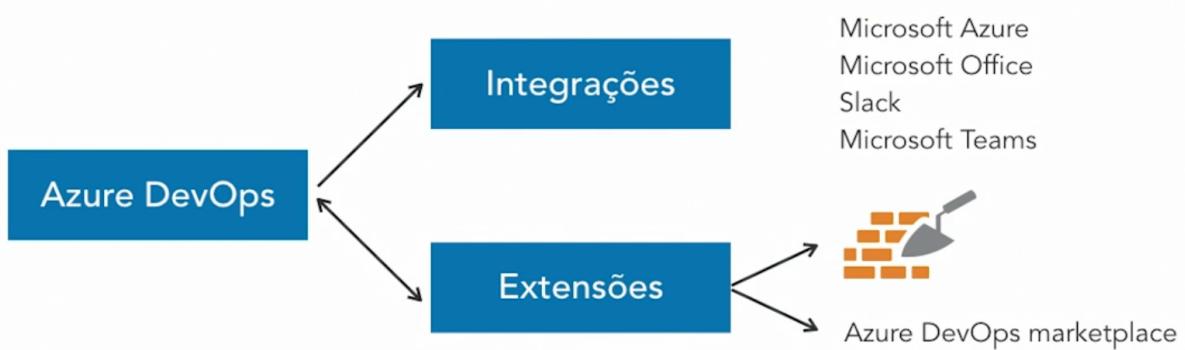
Permite que desenvolvedores, parceiros e equipes individuais criem experiências integradas de primeira classe nos Azure DevOps.

Podendo publicar em um visual studio. Desenvolver sua task ou integração, que após terá seu nome publicado à sua extensão, ou mesmo algum outro tipo de integração.

Integração

Link entre o Azure DevOps e outros aplicativos que permitem que as equipes se inscrevam em eventos e recebam notificações instantâneas para eles.

É um pouco diferente do que conhecemos.



INTEGRAR COM NOTIFICADORES

Podemos trabalhar com o Teams, Slack, entre outras ferramenta de comunicação.
Nele podemos criar subscription. Criar ou pesquisar Work Item

Gustavo Ragghiante

<https://linkedin.com/in/gustavo-ragghi>
gragghi@gmail.com

Skype: gustavo.ragghi