

## Bootcamp IGTI: Arquiteto Cloud Computing

### Desafio

<b>Módulo 4</b>	<b>SDV - Soluções para Desenvolvimento</b>
-----------------	--

### Objetivos

Utilizar os principais recursos para desenvolvimento de aplicações na Azure, exercitando os seguintes conceitos trabalhados neste Módulo (DevOps):

- ✓ **Primeira Maneira:** Acelerar o fluxo dos desenvolvedores (esquerda) para a operação e cliente (direita).
- ✓ **Segunda Maneira:** Rápido feedback em todos os estágios do fluxo de valor (direita para a esquerda).
- ✓ **Terceira Maneira:** Cultura de alta confiança que permite correr riscos e potencializar o aprendizado contínuo.

O aluno deve ter contas gratuitas do AzureDevOps e Azure Services (disponível gratuitamente durante 1 mês) e do GitHub.

### Enunciado

O aluno deverá desenvolver os principais princípios de DevOps e Engenharia de Software aprendidos neste e em todos os outros três módulos, consolidando o aprendizado ao longo do curso.

Neste trabalho prático, o aluno vai utilizar as ferramentas GitHub, Azure DevOps e Azure Services, exercitando a visão do desenvolvedor na abordagem DevOps. Será fornecido de exemplo um código simples, para que seja possível publicá-lo (configurando uma release) continuamente em um Ambiente de Testes e, após a aprovação, automaticamente em outro de Produção, consolidando importantes conceitos de qualidade e segurança ao longo deste

processo de desenvolvimento, tais como: Pipeline, Release, Pull Request e Gestão de Mudanças.

## Atividades

Os alunos deverão desempenhar as seguintes atividades na Primeira Maneira (Acelerar o fluxo dos desenvolvedores (esquerda) para a operação e cliente (direita)):

1. Tornar o trabalho visível e reduzir o tamanho dos lotes e intervalos usando o AzureDevOps.
  - a. Usar o template do Scrum do AzureDevOps, conforme a estrutura proposta no Vídeo 1 para a entrega do Módulo EAD (Épico), Tela Principal (Feature), traduzir para português o Resultado (História), Mudar Texto (Task), realizar testes no ambiente Desenvolvedor (task).
2. Entender a teoria das Restrições e como otimizar o fluxo: remover desperdícios; foco no cliente e limitar o trabalho em andamento, usando os dashboards do Azure DevOps (Métricas de Desenvolvimento).
  - a. Criar um dashboard, uma query e analisar as métricas;
  - b. Customizar o board e ajustar o WIP (Working in Progress) de uma das etapas do board.
3. Reduzir o número de transferências (handoff) - Montagem das etapas do quadro.
  - a. Customizar as etapas do quadro para incorporar um processo que dê mais visibilidade que o default do AzureDevOps.
4. Conhecer infra como código e self-service – Azure Services – conceitos básicos.
  - a. Criar um ambiente de testes e outro de produção no Azure Services;
  - b. Associar o GitHub ao AzureDevOps.
  - c. Criar Pipeline (Build) Integração contínua (usando o arquivo YAML disponível e pronto para esta criação).

- d. Ver como funciona a gerência de configuração deste código (build) no AzureDevOps e no GutHub.
5. Incorporar qualidade na origem (Qualidade desde o início, Testes Automatizados).
- a. Entender a importância dos critérios de ready e done, análise de impacto para a segurança nas histórias e testes, e inserir/consultar estas informações no AzureDevOps. Os testes, por exemplo, são executados durante a CI; critérios de ready, done e segurança são incluídos nas histórias.
6. Entrega e Implantação Contínua.
- a. Configurar uma Release no AzureDevOps para implantação do resultado do CI no ambiente de testes (criado no Azure Services) e, depois, configurar uma aprovação no Ambiente de Testes para ida ao ambiente de produção (criado no Azure Services).

Os alunos deverão desempenhar as seguintes atividades na Segunda Maneira (Rápido feedback em todos os estágios do fluxo de valor-direita para a esquerda):

- 1. Qualidade próxima da fonte (menos aprovações).
  - a. Revisão de Código – Realizar um Pull Request no GitHub e observar a integração no Azure DevOps.

Os alunos deverão desempenhar as seguintes atividades na Terceira Maneira (Cultura de alta confiança que permite correr riscos e potencializar o aprendizado contínuo):

- 1. Cultura de alta confiança que permite correr riscos e potencializar o aprendizado contínuo.
  - b. Como incluir uma retrospectiva no Azure DevOps.
- 2. Difundir conhecimento usando testes automatizados.
  - a. Difundir conhecimento usando o BDD (usar a Wiki e estrutura para incluir o BDD).



