



# Capacitação Tecnológica em Engenharia e Desenvolvimento de Software

D8 – Avaliação

**Professor:** Saulo Santos [ITuring] [saulo.macedosantos@gmail.br](mailto:saulo.macedosantos@gmail.br)

Julho de 2022



# Escopo

- A partir de um projeto base, adicionar as funcionalidades solicitadas e fazer a publicação do sistema utilizando as práticas do DevOps.



# Escopo

- Critérios:
  - Deverá ser criado um projeto na plataforma Azure DevOps (<https://azure.microsoft.com/pt-br/products/devops/>) para reunir o fluxo de trabalho automatizado. O nome desse projeto deverá ser **Books\_NomeAluno1\_NomeAluno2**.
  - Convidar o usuário a seguir para o projeto: [saulo.macedosantos@outlook.com](mailto:saulo.macedosantos@outlook.com).
  - O fluxo de trabalho deve ser organizado no Azure Boards.
  - O código fonte do projeto deve estar armazenado no Azure Repos.
  - O sistema deverá ser publicado na plataforma de serviços em nuvem da Azure (<https://azure.microsoft.com/pt-br/get-started/azure-portal/>) de forma automatizada, ou seja, criando pipelines de integração (CI) e de entrega (CD), usando a ferramenta do DevOps Azure Pipelines.
  - As funcionalidades que deverão ser desenvolvidas são:
    - Listagem de Autores e Livros;
    - Cadastro de Autores e Livros;



# Escopo

- Observações:

- Todas as funcionalidades devem ser testadas (teste automatizado de unidade).
- Criar uma sprint com início em 03/10/2022 e término em 07/10/2022.
- Cada integrante deve se responsabilizar por uma feature completa, inclusive os testes.  
Exemplo: Aluno 1 faz a listagem e o cadastro de autores e o aluno 2 faz a listagem e o cadastro de livros.
- Organize o repositório de modo que cada feature tenha uma branch específica. Exemplo:
  - main > dev > list-books
  - main > dev > create-book
  - main > dev > list-authors
  - main > dev > create-author
- Desta forma, com a pipeline criada, a cada novo push na branch dev, um novo release é gerado com a nova funcionalidade integrada.



# Escopo

- Observações:
  - Código para a task de teste na pipeline de CI:

```
- task: VSTest@2
  displayName: 'Run Tests'
  inputs:
    testAssemblyVer2: |
      **\*Tests*.dll
      !**\*TestAdapter.dll
      !**\obj\**
      !**\bin\**\ref\**
    diagnosticsEnabled: True
```

- Código para a task de criação de artefato na pipeline de CI:

```
- task: PublishBuildArtifacts@1
  inputs:
    PathToPublish: '$(Build.ArtifactStagingDirectory)'
    ArtifactName: 'drop'
    publishLocation: 'Container'
```



# Escopo

- Observações:

- Não esqueça de configurar o arquivo .gitignore no repositório do Azure Repos para evitar que o arquivo appsettings.local.json seja enviado:

A screenshot of a code editor showing the contents of a .gitignore file. The file name '.gitignore' is at the top. Below it, there are two tabs: 'Contents' (selected) and 'Highlight changes'. The code content shows line numbers 363, 364, 365, and 366. Line 364 contains the text '\*\*/appsettings.local.json', which is the pattern used to ignore the local application settings file.

- Configure o serviço de aplicativo para que haja a comunicação com o serviço de banco de dados.
  - Nas configurações do aplicativo, clique em **New application setting** e defina o nome como **ASPNETCORE\_ENVIRONMENT** e o value como **Development**
  - Após, clique em **New connection string** e define o name como **Default**, o type como **SQL Server** e o value com o que foi copiado da cadeia de conexão do banco de dados.



# Entrega

- A entrega deverá ser realizada através do formulário abaixo até o dia **10/10/2022**.
- <https://forms.gle/rU4fFfRj9wsPQAqp6>