

DEVOPS TOOLS & CLOUD COMPUTING ^{1/4}

Primeira opção:

- 01) Containerização da API solicitada na disciplina ADVANCED BUSINESS DEVELOPMENT WITH .NET em Nuvem (até 85 pontos)
- 02) Documentação no GitHub (até 15 pontos)

As seguintes regras solicitadas por essa disciplina serão avaliadas para a execução em nuvem:

- . Apresentar um CRUD (GET, POST, PUT, DELETE) com pelo menos 5 inserts com conteúdo significativo
- . Integração do Banco de dados Oracle
- . Open API Implementada seguindo os padrões para documentação das API's com interface gráfica (Swagger, Redoc ou Scalar)
- . ReadMe do projeto e Github
- . Implementação do Readme do projeto apresentando: Descrição do projeto, Rotas, Instalação, Dockerfile, scripts do Azure CLI

A nota irá variar de acordo com a completude da entrega e, se a entrega não funcionar em nuvem, não poderá ser avaliada, ficando apenas a parte do GitHub para análise e totalização da nota

DEVOPS TOOLS & CLOUD COMPUTING ^{2/4}

Segunda opção:

01) Containerização da API solicitada na disciplina JAVA ADVANCED em Nuvem (até 85 pontos)

02) Documentação no GitHub (até 15 pontos)

As seguintes regras solicitadas por essa disciplina serão avaliadas para a execução em nuvem:

- . Spring Web para criação da API
- . Spring Data JPA para acesso ao banco de dados
- . Banco de dados H2 ou Oracle
- . CRUD completo para pelo menos duas entidades e pelo menos 5 inserts com conteúdo significativo em cada entidade
- . Relacionamento entre entidade
- . O repositório deve conter o código do projeto na raiz, Dockerfile, scripts do Azure CLI
- . O repositório deve ter um readme com a descrição do projeto, nomes dos alunos e instruções para executar

A nota irá variar de acordo com a completude da entrega e, se a entrega não funcionar em nuvem, não poderá ser avaliada, ficando apenas a parte do GitHub para análise e totalização da nota

DEVOPS TOOLS & CLOUD COMPUTING ^{3/4}

Tarefas da disciplina (independente da escolha da tecnologia):

- 01) Provisionar uma Máquina Virtual Linux no Azure via CLI
- 02) Abrir as portas necessárias ao projeto via CLI
- 03) Instalar o Docker na VM criada via SSH
- 04) Executar a entrega de .NET ou Java na VM utilizando Docker
 - Projeto executando em background
 - Imagens otimizadas (ex: python:3.11-slim, node:18-alpine)
 - Rodar aplicação com usuário sem privilégios administrativos
- 05) Acessar externamente e realizar todos os testes
- 06) Desenhe a Arquitetura Macro da sua solução na nuvem (ex: fluxo de usuários, front-end, API, banco de dados, VM, containers, etc.). Utilize ferramentas como Draw.io ou Visual Paradigm (links abaixo) e inclua legendas, rótulos, imagens e setas de fluxo para facilitar a compreensão
- 07) Ao final da entrega delete a Máquina Virtual criada

Obs.: Pode gerar a imagem do Docker em sua máquina de desenvolvimento e subir para o Docker Hub. Assim a equipe somente roda a solução na VM baixando a imagem

DEVOPS TOOLS & CLOUD COMPUTING 4/4

Entrega:

Grave um vídeo demonstrando: A Criação da VM via CLI, a aberturas das portas necessárias via CLI, Instalação do Docker, Funcionamento da aplicação com o Docker e Persistência de dados (**mostrar cada operação executada no banco**)

Entrega final:

Um único arquivo PDF nomeado como: **NomeDoGrupo_Challenge_1Sem_2TDS.pdf**, contendo:

- 01) Uma folha de rosto com o nome da equipe, RM e nome completo dos alunos. Inclua um índice para organização
- 02) Desenho da arquitetura com legenda, fluxos etc
- 03) Link para o repositório no GitHub
- 04) Link para o vídeo no YouTube
- 05) Print da evidência da remoção da VM

Sugestões de ferramentas de desenho

- Azure Diagram Tool (Visual Paradigm)

<https://online.visual-paradigm.com/diagrams/features/azure-architecture-diagram-tool/>

- Draw.io

<https://app.diagrams.net/>