Roteiro
Aula Prática

Infraestrutura Ágil

ROTEIRO DE AULA PRÁTICA
NOME DA DISCIPLINA: **Infraestrutura Ágil**

OBJETIVOS

Definição dos objetivos da aula prática:

- Simular o monitoramento de processo de pipeline de entrega, utilizando o GIT.

INFRAESTRUTURA

Instalações:

GIT.

Materiais de consumo:

Descrição	Quantidade de materiais por procedimento/atividade
Computador	1 por aluno

Software:

Sim (X) Não ( )

Em caso afirmativo, qual? GIT.

Pago ( ) Não Pago (X)

Tipo de Licença: Freeware.

Descrição do software:

O Git é um sistema de controle de versões distribuído, usado principalmente no desenvolvimento de software, mas pode ser usado para registrar o histórico de edições de qualquer tipo de arquivo.

Equipamento de Proteção Individual (EPI):

- NSA

PROCEDIMENTOS PRÁTICOS

Realizar o monitoramento de processo de pipeline de entrega, utilizando o GIT.

Atividade proposta:

- Entender como funciona o script para realização da chamada Integração Contínua.
- Criar um relatório no final da atividade.

Procedimentos para a realização da atividade:

Nesta atividade iremos criar nosso primeiro script que irá possibilitar a realização de uma Integração Contínua. De maneira geral, a grande maioria das ferramentas possuem um processo semelhante ao do GitLab CI/CD. No caso do GitLab CI/CD o pipeline é definido dentro de um arquivo denominado .gitlab-ci.yml, que segue o formato do YAML, que nada mais é do que uma linguagem de marcação. Este arquivo define a ordem em que se dará a execução do pipeline.

Vamos apresentar os conceitos essenciais para a compreensão do pipeline que desejamos para a construção da nossa aplicação de forma automática e posterior entrega da mesma no ambiente de produção.

Um arquivo .gitlab-ci.yml é formado, basicamente, por um conjunto de jobs, na terminologia do GitLab CI/CD. Jobs são os elementos mais básicos dentro do arquivo .gitlab-ci.yml.

Conforme a documentação oficial, jobs são:

- Definido com restrições informando em que condições devem ser executados;
- Elementos de nível superior com um nome arbitrário e que devem conter pelo menos a cláusula script.
- Ilimitados dentro de um arquivo .gitlab-ci.yml.

Vamos dar início a definição do pipeline para o projeto de uma Loja Virtual.

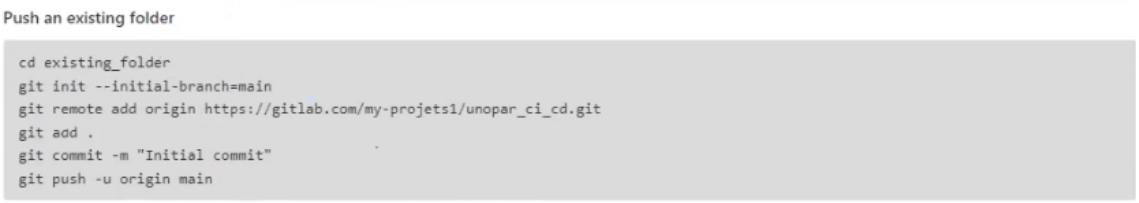
Será necessário criar uma conta no gitlab: https://gitlab.com/users/ e instalar o Git: https://git-scm.com/downloads.

Também iremos criar uma conta no hub.docker: https://hub.docker.com/ para o container.

O arquivo “devops-master”, será descompactado numa pasta em seu computador. E para o projeto serão utilizados os arquivos abaixo:

Nome	Status	Data de modificação	Tipo	Tamanho
.gitlab-ci.yml	✓	16/10/2022 21:07	Arquivo YML	2 KB
Dockerfile	✓	16/10/2022 21:07	Arquivo	1 KB
notificacaoFalha.sh	✓	16/10/2022 20:52	Shell Script	1 KB
notificacaoSucesso.sh	✓	16/10/2022 20:52	Shell Script	1 KB
plot_simples_grafico.py	✓	16/10/2022 21:06	Python File	2 KB
relogio.py	✓	16/10/2022 20:52	Python File	1 KB

Estando no repositório do projeto UNOPAR_CI_CD, você irá "puxar" os arquivos para Gitlab, seguindo a estrutura da imagem abaixo:



Esse script de Integração Contínua faz uso da imagem, para construir a aplicação. Por padrão, as imagens são baixadas do Docker Hub mas é possível alterar essa configuração se desejado. Neste caso, faremos uso do Docker Hub padrão para a busca e registro de imagens.

Ao terminar da execução, o job encerrando com sucesso, define que o pipeline aprovou último commit realizado. Se houver falha na execução do script, o job é reprovado e, consequentemente, o pipeline acusa uma falha.

Checklist:

- Instalar o sistema GIT.
- Simular se houve sucesso ou falha na execução do job.

RESULTADOS

Resultados da aula prática:

Elaborar um relatório que deverá conter introdução, métodos, resultados e conclusão sobre o assunto desenvolvido em aula prática, para compreender o funcionamento essenciais do pipeline para a construção da aplicação de forma automática.