

Pontifícia Universidade Católica de Minas Gerais

Praça da Liberdade

Engenharia de Software

Prof. Johnatan A Oliveira

Trabalho Individual de Gestão de Configuração de Software 2023/1

Versão JAVASCRIPT

10 Pontos

1 Descrição

Os conhecimentos de Gestão de Configuração de Software são fundamentais no ciclo de vida de um produto de software. As técnicas para a gestão vão desde o controle de versão, automação de build e de configuração de ambiente, testes automatizados, isolamento do ambiente até o deploy do sistema. Este ciclo são atualmente integrados em um pipeline de DevOps com as etapas de Integração Contínua (CI) e Deploy Contínuo (CD) implementadas e automatizada.

Para exercitar estes conhecimentos, neste trabalho, você deverá aplicar os conceitos estudados ao longo da disciplina no produto de software contido neste repositório.

O sistema se trata de uma aplicação em Django e React que gerencia uma biblioteca. Para executar a aplicação em sua máquina, basta seguir o passo-a-passo descrito nas pastas do Backend e Frontend arquivo. Link do projeto: [PUC-GCES-JS](#)

2 Sobre a Aplicação

A aplicação é um simples sistema de gerenciamento de uma biblioteca. Porém, o foco do trabalho é na automação da build, testes, containerização e configuração dos pipelines de CI/CD.

3 Etapas do Trabalho

O trabalho deve ser elaborado por etapas. Cada uma das etapas deve ser realizada em um commit separado com o resultado funcional desta etapa. As etapas de 1 a 3 (VEJA A TABELA 1) são relacionadas ao isolamento do ambiente utilizando a ferramenta Docker e Docker Compose. Neste sentido o tutorial abaixo cobre os conceitos fundamentais para o uso destas tecnologias.

3.1 Tutoriais

As etapas de 4 a 6 (VEJA A TABELA 1) são relacionadas à configuração do pipeline de CI e CD.

Links

Clique: [Tutorial de Docker](#)

Clique: [Tutorial CI - Gitlab](#)

4 Containerização do Banco

A versão inicial do sistema contém uma aplicação Django no backend cujo funcionamento requer uma instalação de um banco de dados Postgres. A primeira etapa do trabalho é de configurar um container somente para o banco de dados com as credenciais especificadas na descrição da aplicação e testar o funcionamento do mesmo.

5 Containerização da aplicação + banco

Nesta segunda etapa, tanto o backend quanto o banco de dados deverão estar funcionando em containers individuais. Deverá ser utilizado um orquestrador (Docker Compose) para gerenciar comunicação entre os containers além do uso de credenciais, networks, volumes, entre outras configurações necessárias para a correta execução da aplicação.

6 Configuração do Frontend

A terceira etapa do trabalho consiste em dockerizar o frontend que é uma aplicação em React seguindo as instruções do README e incluindo a mesma no Docker Compose.

7 Servindo a aplicação com o Nginx

A última etapa de configuração do ambiente é a de servir a aplicação através do Nginx. Neste caso, será necessário configurar o Nginx em um container e adicioná-lo ao Docker Compose.

Clique: [Referência 1](#)

Clique: [Referência 2](#)

8 Integração Contínua (CI)

Para a realização desta etapa, a aplicação já deverá ter seu ambiente completamente containerizado. Deverá ser utilizada uma ferramenta de Integração Contínua para garantir o build, os testes e os deploy para o [Docker Hub](#) dos serviços principais. Esta etapa do trabalho poderá ser realizada utilizando os ambientes de CI do GitLab-CI ou Github Actions. Requisitos da configuração da Integração Contínua (Gitlab ou Github) incluem:

- Build
- Test
- Lint

9 Deploy Contínuo (CD)

A etapa final do trabalho deverá ser realizada à partir do deploy automático da aplicação que deve ser realizado após passar sem erros em todas as etapas de CI.

10 Avaliação

A avaliação do trabalho será feita à partir da correta implementação de cada etapa 1 a 6 (TABELA 1). A avaliação será feita de maneira quantitativa (se foi realizado a implementação + documentação), e qualitativa (como foi implementado, entendimento dos conceitos na prática, complexidade da solução). Para isso, faça os commits atômicos, bem documentados, completos a fim de facilitar o entendimento e avaliação do seu trabalho. ***Lembrando o trabalho é individual.***

10.1 Observações

1. A data final de entrega do trabalho é o dia 01/06/2023 até 23:59 horas.
2. O trabalho deve ser desenvolvido em um repositório PESSOAL e PRIVADO que deverá ser tornado público somente após a data de entrega do trabalho (no dia 02/06/2023);
3. Cada etapa do trabalho deverá ser entregue em commits progressivos (podendo ser mais de um commit por etapa);
4. Os commits devem estar espaçados em dias ao longo do desenvolvimento do trabalho. Commits feitos todos juntos na data de entrega serão descontados da nota final.

Table 1: Pesos para a pontuação

Item	Peso
1. Containerização do Banco	1.0
2. Containerização da aplicação + Banco	1.5
3. Containerização da Aplicação + Banco + Front	1.5
4. Containerização da Aplicação + Banco + Front + Nginx	1.5
5. Integração Contínua (Build, Test, Lint)	3.0
6. Deploy Contínuo	1.5

11 ATENÇÃO

Caso exista divergência das datas entre o Canvas e o PDF. A data correta será aquela que está no CANVAS.

12 Referências

O material adotado na disciplina é com base na disciplina dos professores: Renato Coral e Carla Rocha. Ambos são da Universidade de Brasília - UnB.