

Pontifícia Universidade Católica de Minas Gerais

Praça da Liberdade

Engenharia de Software

Prof. Johnatan Oliviera

Trabalho individual de Gestão de Configuração de Software 2023-1

Versão PYTHON

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 biblioteca Python para executar pipelines de dados de forma customizável em bancos de dados.

Link do projeto: PUC-GCES-JS

2 Sobre a Aplicação

A biblioteca desenvolvida auxilia desenvolvedores a explorar os dados com funções essenciais para a identificação de outliers e anomalias e uma interface que auxilia a visualizar as informações conforme o arquivo de configuração. A biblioteca recebe um arquivo yaml com as configurações de cada etapa do pipeline de dados, e do endereço do banco de dados. Após a execução do banco de dados, o banco de dados é atualizado com os resultados da análise e os resultados podem ser visualizados por meio de dashboards no metabase.

3 Etapas do Trabalho

O trabalho deve ser elaborado através etapas. Cada uma das etapas deve ser realizada em um commit separado com o resultado funcional desta etapa. As etapas de 1 a 3 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 Tutotiais

Links

Clique: Tutorial de Docker

Clique: Tutorial CI - Gitlab

4 Containerização do Banco

A versão inicial do sistema contém o metabase no backend cujo funcionamento requer uma instalação de um banco de dados Mongo. 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 + metabase

Nesta etapa, tanto a aplicação Python quanto o metabase/banco 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 Gestão de dependencias e pacotes python

Configurar o gerenciador de dependencias e pacotes python, o poetry, para gerar um pacote pip da solução. Publicar a biblioteca <https://python-poetry.org>

7 Documentação automatizada

Gerar a documentação da biblioteca de forma automatizada utilizando o doxygen para gerar informacoes da biblioteca e o sphinx para criar documentação <https://www.sphinx-doc.org>

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 o deploy para o [hyperlinkhttps://pypi.orghttps://pypi.org](https://pypi.org). 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 (Poetry) Test - unitários Lint - Documentação (sphinx)

9 Avaliação

A avaliação do trabalho será feita à partir da correta implementação de cada etapa. 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.

9.1 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 (pendendo 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 biblioteca + Banco | 1.5 |
| 3. Publicação da biblioteca | 1.5 |
| 4. Documentação automatiza | 1.5 |
| 5. Integração Contínua (Build, Test, Lint, documentacao) | 3.0 |
| 6. Deploy Contínuo | 1.5 |

10 ATENÇÃO

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

11 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.