**Documentação de projeto**

1. **Desenho de arquitetura:**

Diagrama

Descrição gerada automaticamente

1. **Origem**

https://randomuser.me/api/?results=10

1. **Requisitos do Sistema**
2. Armazenamento dos dados brutos em diretório específico.
3. Armazenamento dos dados saneados em tabela mysql.
4. Tratamentos mínimos dos dados:
   * Tratamento de tipos
   * Tratamento de nomes
   * Padronização dos dados: remoção dos caracteres especiais e transformação das strings para lowercase() (minúscula). Aplicar tratamento em uma coluna qualquer (que for string), basta especificar no metadado.
5. Não é necessário fazer seleção dos dados como foi feito em aula, fica a escolha do grupo.
6. Arquivo de metadado com ao menos nome\_original, nome\_renomeado, tipo e a especificação da coluna do tratamento do item c.
7. É necessário inserir o log de informação para cada etapa do processo de ingestão e preparação).
8. É necessário inserir tratamento de exceção para cada etapa do processo de ingestão e preparação).
9. Dockerfile:
   * Mysql com a criação da tabela
   * Python para a aplicação de ingestão dos dados
10. Scripts python com docstring
11. Docker-compose para levantar a aplicação python e o mysql
12. Implementação de um workflow gitactions:
    * Para validação de escrita de código python com pylint. Nota deve ser superior a 5.
    * Salvar a imagem da aplicação python no dockerhub
13. **Etapas do processo**
    1. Ingestão dos dados da API (10 linhas)
    2. Armazenamento dos dados brutos em diretório específico
    3. Tratamento dos dados conforme o item 3.a
    4. Armazenamento dos dados saneados em tabela mysql
    5. Workflow
14. **Formato de entrega**
    1. Link do github (TODOS OS MEMBROS DA EQUIPE DEVEM MANDAR O LINK DO GIT) contendo ao meno:
       1. Scripts python
       2. Metadados
       3. Dockerfile
       4. Docker-compose
       5. Workflow actions
       6. Readme explicando o projeto
    2. Vídeo de demonstração ou prints das seguintes etapas:
       1. Execução do docker-compose (não se esqueçam que na primeira execução o python tende a falhar devido ao time de criação da tabela no mysql. Se ocorrer, pare o container e rode novamente)
       2. Logs do processo de ingestão sendo printados como logs do docker
       3. Select \* from <tabela> através do comando exec -it do docker
       4. Printar o conteúdo da tabela através de um jupyter notebook (localhost)
       5. Git push e execução da esteira
       6. Mostrar os passos do pylint e de deploy no Dockerhub
       7. Mostrar a imagem no Dockerhub

O grupo deve entregar um documento word/pdf com:

* Nome dos integrantes do grupo
* Link do github de cada membro do time
* Prints das etapas (caso o grupo prefira prints no lugar do vídeo)

1. **Combinados**
   1. Grupo de até 5 pessoas
   2. Data da entrega: 28/04/2024
   3. Não será aceito projetos com credencial exposta. Usem cofre do actions.