Plano de Trabalho (Realizado)

EXTRAÇÃO, CARGA, TRANSFORMAÇÃO E CONTROLE DE DADOS DE REDE SOCIAL – TWITTER – PROCESSO SELETIVO ENGENHEIRO DE DADOS

Controle de versão: ***v2.0***

Nome	Evento	Número de páginas	Data
Guilherme Ditzel Patriota	Criação da v1.0	5	21/07/2021
Guilherme Ditzel Patriota	Atualização para v2.0	5	25/07/2021

DESCRIÇÃO DO PLANO DE TRABALHO REALIZADO

Desenvolvida aplicação para coleta e armazenamento dos dados para consumo de postagens em tempo real do Twitter, em português, com as palavras “COVID” e “Saúde”.

GESTÃO DE ARQUIVOS, VERSIONAMENTO E ESTRUTURA DE PASTAS

Para garantir a qualidade dos processos deste projeto foram adotados alguns padrões durante todo o desenvolvimento.

Gestão de arquivos

A gestão de arquivos de desenvolvimento está inteiramente em git, com uso de repositório em nuvem github (https://github.com/guipatriota/Pipeline_COVID_SAUDE_BR-HiTechnology).

O uso do github permitiu o fácil compartilhamento e controle de atualizações dos arquivos durante o desenvolvimento e futuramente contribuições da comunidade de software livre serão bem vindas, tendo em vista que este repositório será de acesso público.

O projeto está sob a licença MIT, o que permite o uso livre deste conteúdo por outras pessoas da forma que bem entenderem, desde que haja citação ao autor do mesmo.

De forma local, o software VSCode foi usado para gerir as pastas, arquivos e commits dos mesmo para o servidor do github, bem como para interface com o docker e dockerhub.

Versionamento de código

Para o fluxo de trabalho e seu versionamento foi utilizado o gitflow, modelo de ramificações para melhor organização das branchs do projeto, sendo inicialmente compostas pelas seguintes ramificações:

- master
- develop

Para o versionamento, a TAG v0.1 foi usada na branch master para a parte inicial do projeto, que atingirá a v1.0 ao final deste plano de trabalho.

Futuras ramificações de features e outras poderão ser incluídas em uma segunda etapa deste projeto, mas não farão parte do escopo deste plano de trabalho.

Estrutura de pastas e padrão de nomenclaturas

Este projeto utiliza o padrão PEP-8 para nomenclatura de arquivos, pastas, classes, funções e variáveis:

- Nomes de pastas: curto, letras minúsculas e sem sublinhado (ex.: pastadetrabalho)
- Nomes de variáveis, funções e arquivos: curto, letras minúsculas separadas por sublinhado (ex.: coleta_de_dados)
    - Exceção: README.md e LICENSE.md
- Nomes de classes: curto, primeira letra de cada palavra em maiúsculo e sem espaços (ex.: ColetaTweets)

O projeto conterá a seguinte estrutura de pastas inicial:

- master – branch principal do projeto. Pasta raiz.
    - /docs – Pasta com a documentação do projeto e manual
        - /projeto – Documentação referente ao plano de trabalho
        - /api – Documentação de uso da API REST para consumo dos dados coletados
        - /arquitetura – Documentação da arquitetura do Banco de Dados
    - /hitweets – Pasta com os arquivos da aplicação
        - /db – Pasta com arquivos de criação e gestão do banco de dados
        - /colect – Pasta com os arquivos de coleta dos dados
            - /data – Pasta com arquivos JSON temporários de coletas em andamento e finalizadas
                - /_old – Pasta com os arquivos json originais recebidos do Twitter
                - /datalake – Pasta com os arquivos limpos para uso no banco de dados.
        - /api – Pasta com a API para consumo dos dados coletados e sumarizados
        - /log – Pasta com arquivos JSON e TXT de log.
        - /transform – Pasta com o script de limpeza da base de dados
    - /tests
    - README.md – Documentação inicial dor projeto e da aplicação
    - LICENSE.md – Licença sob a qual este projeto foi desenvolvido e disponibilizado.
    - .gitignore – Arquivo com pastas e arquivos que não devem ser enviados para a nuvem do github
    - requirements.txt – Arquivo com dependências do projeto

COLETA DE DADOS

São coletados no mínimo 2000 (postagens na rede social Twitter) em tempo real a cada 30 minutos (hora cheia hh:00 e hh:30).

A coleta utiliza o "*filtered stream*" da API Academic Research do Twitter com as seguintes regras:

{"value": "COVID lang:pt", "tag": "Covid rule"}

{"value": "Saúde lang:pt", "tag": "Saúde rule"}

Esta API permite uma coleta máxima de 10 milhões de tweets por mês, com 1000 regras por aplicativo e 50 solicitações a cada 15 minutos

Foi optado pelo uso da API Academic Research e não standard por conta de acesso previamente existente.

Como as postagens a serem coletadas são as de acesso público e o acesso é apenas de leitura na base de dados da rede social, sendo utilizado o método de autenticação OAuth 2.

A linguagem a ser utilizada para a criação da aplicação de coleta foi o Python, em conjunto com a biblioteca Requests.

A coleta de dados é recebida com as seguintes informações:

Nome do campo	Tipo do dado	Descrição do campo	Parâmetro da query
id	String	Identificador único do tweet	Padrão
text	String	Conteúdo em texto da postagem	tweet.fields=text
created_at	Date (ISSO 8601)	Data da postagem	tweet.fields=created_at
author_id	String	Identificador único do usuário autor da publicação	expansions=author_id

Estes dados coletados permitem o fornecimento de dois tipos de informação:

1. Quantidade total de postagens agrupadas por horário do dia na soma total de dados coletados, fornecida a cada 30 minutos
2. Quantidade total de postagens para cada uma das regras (regra contendo COVID e regra contendo Saúde), também a cada 30 minutos e agrupada por horário do dia

Os campos “text” e “author_id” não estão disponíveis para uso, porém poderão ser adicionados futuramente, caso necessário.

Os dados coletados estão disponibilizados em arquivo JSON, que pode ser consumido pelo banco de dados

API A SER DESENVOLVIDA PARA CONSUMO DESTES DADOS

Até o momento não foi possível finalizar o desenvolvimento de uma API para consumo dos dados.

DOCUMENTAÇÃO

A documentação possui as seguintes partes:

- Documentação do projeto e plano de trabalho
- Documentação da aplicação em HTML

ÉTICA E CONFORMIDADE LEGAL

Pelo acordo de restrição de uso dos dados com o Twitter para o uso da API Academic Research, não se deve apresentar os dados de forma a identificar individualmente cada usuário. Por este motivo o nome dos usuários não é fornecido pela aplicação.

Apesar desta restrição, é possível a coleta de informações pessoais para fins estatísticos, desde que nomes individuais não sejam apresentados em resultados finais ou em apresentações públicas dos dados coletados.

Tendo em vista esta necessidade de segurança dos dados e a lei geral de proteção de dados brasileira, o banco de dados deverá possuir restrição de acesso aos dados individuais e apenas dados sumarizados deverão ser fornecidos do mesmo.

Os dados coletados não poderão ser utilizados para nenhum outro fim se não o do escopo deste plano de trabalho.

Para a apresentação final deste projeto foi usado o perfil Academic Research sem grandes dificuldades.

O APP criado no Twitter consumo da API para este projeto ficará disponível até dia 31/07/2021. Após esta data, os dados não poderão mais ser coletados com o uso da chave disponibilizada nos arquivos deste projeto e deverão ser substituídas por chaves próprias de cada desenvolvedor ou empresa.

Para as demais questões legais, o projeto estará sob a licença MIT descrita anteriormente.

ARMAZENAMENTO E BANCO DE DADOS

Até o momento não foi finalizada a criação do banco de dados para este projeto.

UTILIZAÇÃO E DISTRIBUIÇÃO

Duas imagens docker foram criadas, porém ainda não foram finalizadas, sendo uma para o banco de dados e outra para a aplicação de coleta.

RECURSOS EXTRAS

Não foram incluídos recursos extras até o momento.

CONSIDERAÇÕES FINAIS DO PROJETO

O plano de trabalho previsto descria as etapas e funcionalidades previstas para este projeto e que deveriam ser entregues até dia 25/07/2021, entretanto não foi possível a finalização do mesmo por ter sido um projeto ambicioso para execução e cinco dias, o que acabou não se concretizando.

Cronograma do projeto

ETAPAS

00. DOCUMENTAÇÃO – Incompleto.

01. COLETA DE DADOS – Completo.

02. DB ELASTICSEARCH – Não iniciado.

03. API REST DJANGO – Não iniciado.

04. DASHBOARD PROMETHEUS – Não iniciado.

05. DOCKER DEPLOY – Incompleto.

06. FINALIZAÇÃO – Incompleto.

A SEGUIR, É APRESENTADA A ESTRUTURA ETL PENSADA PARA ESTE PROJETO.

Arquitetura do pipeline – ELT

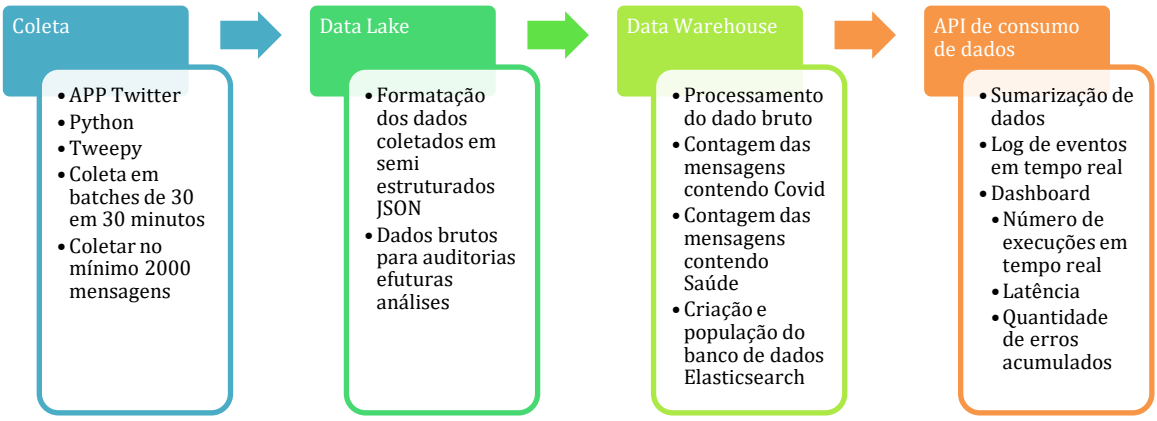

Figura 1 - Diagrama da arquitetura do pipeline da coleta ao consumo do dado