# Plano de Trabalho (Previsto)

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

*Controle de versão:* ***v1.0***

| Nome | Evento | Número de páginas | Data |
|---|---|---|---|
| Guilherme Ditzel Patriota | Criação da v1.0 | 5 | 21/07/2021 |

# DESCRIÇÃO DO PLANO

Desenvolvimento de aplicação para coleta, armazenamento e fornecimento 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 serão adotados alguns padrões a serem seguidos durante todo o desenvolvimento.

## Gestão de arquivos

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

O uso do github permitirá o fácil compartilhamento e controle de atualizações dos arquivos durante o desenvolvimento e possíveis contribuições futuras da comunidade de software livre, tendo em vista que este repositório será de acesso público.

O projeto estará sob a licença MIT, o que permitirá 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 será usado para gerir as pastas, arquivos e commits dos mesmo para o servidor do github.

## Versionamento de código

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

- master
- develop
- release

Para o versionamento, a TAG v0.1 será 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 usará 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
        - /coleta – Pasta com os arquivos de coleta dos dados
            - /data – Pasta com arquivos JSON temporários de coletas em andamento e finalizadas
        - /api – Pasta com a API para consumo dos dados coletados e sumarizados
        - /log – Pasta com arquivos JSON e TXT de log.
    - /testes
    - 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

Serão coletados no mínimo 2000 (dois mil) e máximo de 2500 tweets (postagens na rede social Twitter) em tempo real a cada 30 minutos (hora cheia hh:00 e hh:30).

A coleta utilizará 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 serão as de acesso público e o acesso será apenas de leitura na base de dados da rede social, será utilizado o método de autenticação OAuth 2.

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

A coleta de dados deverá retornar 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 permitirão 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 serão usados inicialmente neste projeto, porém são informações que poderão se mostrar relevantes em uma segunda etapa. Sua coleta precoce permitirá a avaliação da necessidade e testes preliminares com os dados já coletados, permitindo agilidade no processo interno da empresa.

Caso sejam necessárias alterações nos tipos de metadados coletados, será de fácil inclusão ou exclusão com simples alteração dos campos da request feita à API do Twitter.

Os dados coletados serão disponibilizados em arquivo JSON temporário, que será posteriormente consumido para geração e incremento do banco de dados

# DOCUMENTAÇÃO E METADADOS DA API CRIADA

A documentação será dividida em três partes:

- Documentação do projeto e plano de trabalho
- Uso da API REST para consumo dos dados
- Documentação da arquitetura de dados utilizada

Os parâmetros da query para consulta ao banco de dados serão:

- agrupamento
    - agrupamento=geral – retornará a quantidade geral de postagens agrupadas por horário do dia em grupos de 30 em 30 minutos (valor padrão caso nada seja enviado)
    - agrupamento=covid – retornará a quantidade de postagens agrupadas com a palavra “COVID” por horário do dia em grupos de 30 em 30 minutos
    - agrupamento=saude – retornará a quantidade de postagens com a palavra “Saúde” por horário do dia em grupos de 30 em 30 minutos

- periodo
    - periodo=hora – retornará dados da última hora coletada
    - periodo=dia – retornará todos os dados do dia atual
    - periodo=semana - retornará todos os dados referentes aos últimos 7 dias
    - periodo=mês – retornará os dados dos últimos 30 dias

Os metadados retornados pela API criada serão:

- id – Identificador único do request
- quantidade – Quantidade total de postagens agrupadas por horário do dia (em grupos de 30 minutos)
- agrupamento – Identificação de qual regra de agrupamento os dados de quantidade se referem:
    - COVID – Postagens que contenham COVID
    - SAÚDE – Postagens que contenham Saúde
    - GERAL – Todas as postagens
- time_stamp – Data e hora do agrupamento

## ÉTICA E CONFORMIDADE LEGAL

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

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 deverá ser avaliado se o perfil Academic Research realmente é o mais adequado para a entrega ao cliente final ou se o perfil Standard deverá ser utilizado em seu lugar, sob pena de não poder gerar o relatório mensal anterior ao início das coletas, devido à característica de coleta limitada ao máximo de 7 dias de postagens, deste perfil.

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

O processo de coleta de dados será feito automaticamente a cada 30 minutos, iniciando-se em hora cheia. Cada coleta será armazenada em formato de arquivo JSON para então ser carregado ao banco de dados.

O banco de dados escolhido para este projeto será não relacional (NoSQL) do tipo Elasticsearch, por oferecer facilidade para futuras alterações dos dados e seus metadados e ainda fornece as ferramentas para logging necessárias ao projeto.

Este banco de dados terá cada entrada como um documento e este será uma postagem única e sua criação será realizada a partir do primeiro arquivo JSON importado nele, com interpretação dos tipos de forma automática por ele, uma das vantagens de se usar o Elasticserach como banco de dados.

Devido à sua natureza de orientação à documento o banco de dados baseado em Elasticsearch permite rápido processo de armazenamento e busca de informações, mas não para alteração de dados específicos em cada documento. Como alterações não fazem parte do escopo deste projeto, optou-se pela eficiência na recuperação dos dados e segurança.

# UTILIZAÇÃO E DISTRIBUIÇÃO

O ambiente de execução do banco de dados e de todos os processos desta aplicação serão feitos em uma imagem Docker personalizada em um dockerfile. Esta imagem poderá ser testada localmente e será disponibilizada tanto nos arquivos do projeto quanto no Dockerhub.

Tendo em vista a natureza do projeto, não serão feitas considerações sobre utilização de memória, armazenamento nem processamento para o servidor que rodará a imagem docker, entretanto para uma utilização comercial de u projeto similar, a rápida escalada no tamanho deste banco de dados deve ser calculada e prevista, assim como a manipulação e destruição dos dados deverá ser projetada e programada como fim do processo para garantir segurança e utilização por longo período de tempo.

# RECURSOS EXTRAS

Ao final deste projeto serão acessíveis aos usuários 3 tipos de informações:

- Uso da API para consumo dos dados sumarizados do banco de dados
- Uso da API para recuperação de informações em tempo real de log de eventos ocorridos na execução da API, como erros, avisos, debug e outras informações.
- Dashboard com 3 informações referentes à execução da API:
    - Quantidade de execução em tempo real da API
    - Latência/tempo de execução da API
    - Quantidade de erros cumulativos da API até o momento

Para a criação deste dashboard será utilizada a ferramenta Prometheus, pela sua fácil integração com o banco de dados escolhido e com o docker a ser criado.

# CONSIDERAÇÕES DE INÍCIO DO PROJETO

Este plano de trabalho previsto descreve as etapas e funcionalidades previstas para este projeto e deverá ser entregue até dia 21/07/2021 às 09h07min da manhã, quando se encerram as 24h de prazo recebido para tal.

Todas as etapas descritas neste documento deverão ser implementadas e entregues até dia 25/07/2021.

Além da aplicação entregue, este documento deverá ser atualizado com as funcionalidades realmente implementadas e ferramentas utilizadas, gerando então o “plano de trabalho realizado”.

Todo a aplicação, bem como sua documentação deverão ser apresentadas em entrevista com o cliente ao término deste prazo.