# Teste para Data Engineering

**Instruções**

1. Leia esse documento antes de iniciar as atividades.

2. Qualquer dúvida enviar um email para dataengineering@hitechnologies.com.br com o título “Desafio - Engenharia de Dados” e responderemos assim que possível.

2. Você tem 1 dia para entregar o plano de trabalho previsto (item 9), após finalização do trabalho atualizar o plano com a visão do trabalho realizado antes da entrevista. Enviar o plano de trabalho para o email da instrução 2.

3. Você tem até 7 dias corridos para concluir as atividades aqui solicitadas e apresentar o resultado na entrevista (O Resultado deverá ser enviado após os 7 dias, independente do ponto do desenvolvimento);

- Caso não consiga concluir todas as atividades, por favor entregue o que foi feito até a data solicitada.

4. Fique à vontade para utilizar tecnologias, frameworks e técnicas não citadas nas atividades.

5. Recomendamos a utilização do docker (http://www.docker.com) para montagem do ambiente;

- Caso opte pela utilização do docker, publique os Dockerfiles no repositório do projeto ou deixe a imagem publicada no Dockerhub.

**Atividades**

1. Elabore um plano de trabalho.

2. Crie uma aplicação na linguagem que desejar para coletar postagens do Twitter em tempo real utilizando o “filtered stream” da api do Twitter com as seguintes regras: `{"value": "COVID lang:pt", "tag": "Covid rule"},` `{"value": "Saúde lang:pt", "tag": "Saúde rule"}`
a. É interessante salvar o horário de cada tweet para a atividade 5.
b. Quanto mais tweets melhor, mas fica a seu critério quando parar de coletar os dados.(mínimo 2000 tweets)

3. Modele e implemente uma base de dados para armazenar as informações. Os critérios de qual banco utilizar ficam sob sua responsabilidade.

4. Colete e armazene as mensagens em no máximo 30 minutos na base de dados.

5. Utilizando uma linguagem de sua preferência, sumarize e grave os dados em no máximo a cada 30 minutos para conseguir listar as informações:
a. Qual o total de postagens, agrupadas por hora do dia (independentemente da regra utilizada)?
b. Qual o total de postagens para cada uma das regras?

6. Crie uma API REST que permita o consumo dos dois itens anteriores. A Api deverá expor métricas de execução.

7. **Muito importante 1:** Utilizando uma ferramenta de logging (exemplos: Elastic Search, Splunk, Graylog ou similar), crie uma query que mostre em tempo real os eventos que acontecem na execução da api criada no item 6, exemplos (Warning, Erro, Debug, Info, etc). Importante ter ao menos uma situação de execução com erro.

8. **Opcional 1:** Utilizando uma ferramenta de métricas (exemplos: Prometheus, Zabbix, cloudwatch ou similar), crie 3 dashboards que mostre em tempo real a quantidade de execução, a latência (tempo de execução) e quantidade de erros da api criada no item 6. Importante ter ao menos uma situação de execução com erro.

9. Plano de Trabalho (previsto e realizado)

a. Enviar o plano de trabalho realizado no mesmo email da instrução 2, e ele será discutido na entrevista após a entrega do projeto.

10. Publique o projeto no Github e documentar em um README.md os itens abaixo:

a. Documentação do projeto

b. Documentação das APIs

c. Documentação de arquitetura

d. Documentação de como podemos subir uma cópia deste ambiente localmente

Avaliação:

O que importa para nós é aprender como você escreve código e documentação, o que você considera como código limpo e como você geralmente aborda o problema dados os requisitos limitados. Para nós, é mais importante ter um projeto compreensível do que um algoritmo complexo.

Obs.: Para todos os itens iremos considerar a documentação como parte da entrega.

**Importante: Trazer o case funcionando no dia da entrevista.**