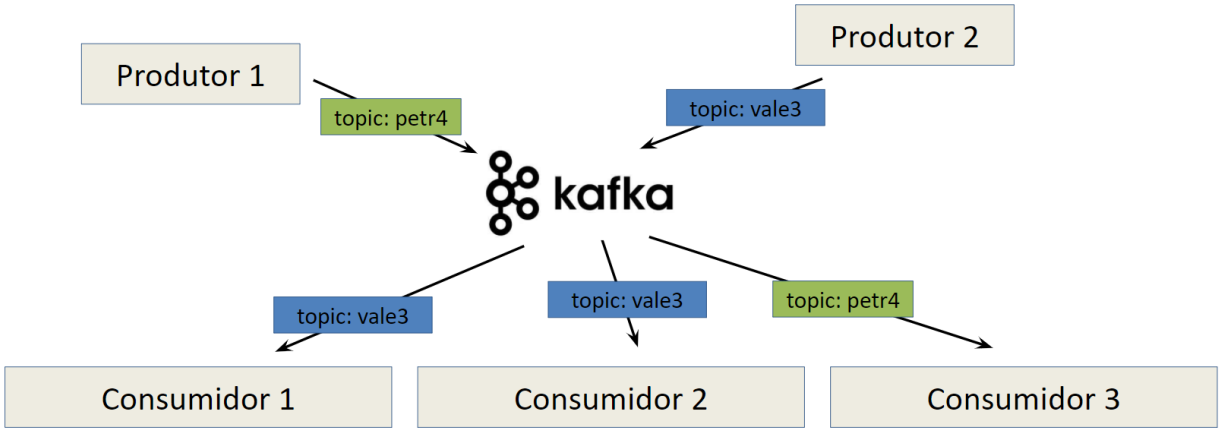
	CENTRO UNIVERSITÁRIO 7 DE SETEMBRO		
	Curso: Especialização em Ciência de Dados		
	Disciplina: Streaming de Dados em Tempo Real	**Professor:** Felipe Timbó	**Data de entrega:** 09/09/2020

Tarefa Final

1) Sobre streaming de dados em tempo real, disserte sobre suas características, o conceito de tempo real, exemplos e principais desafios quando se lida com esse tema. (0,5 pontos)

2) Utilizando os **comandos** (CLI) do Apache KAFKA, crie o seguinte esquema de troca de mensagens via streaming de dados: (1,0 ponto)



Obs. Você deve demonstrar **toda** a sequência de comandos, incluindo a inicialização dos serviços, criação dos tópicos, produção e consumo das mensagens.

3) Considere o script em python *producer_consumer.py*, disponível em www.lia.ufc.br/~timbo/streaming/producer_consumer.py responsável por produzir streaming de dados utilizando Kafka. Altere o código para contemplar os seguintes itens: (2,0 pontos)

   a) No consumidor (classe *Consumer*) imprima apenas as tuplas que possuem valores de IMC acima de 35 (IMC = peso/altura²)

   b) Crie um novo produtor (classe *Producer2*, por exemplo) o qual gera um streaming contendo "nomes" e "salários" aleatórios no intervalo fixo de 4 segundos. Os nomes aleatórios podem ser gerados pela biblioteca *Faker* utilizada no curso e os salários devem estar no intervalo entre R$ 1.000,00 e R$ 3.000,00.

   c) Crie um novo consumidor (classe *Consumer2*, por exemplo) o qual irá consumir os dados do novo produtor criado e imprimir o valor de cada tupla.

   d) Altere o consumidor recém criado para imprimir **apenas o nome** das pessoas que recebem salários maiores que R$ 2.000,00

   e) Aumente a frequência de geração das tuplas para 2 segundos no Produtor 1 (classe *Producer*)

   f) Gere dados simultâneos de dois Produtores da classe Producer e dois produtores da classe *Producer2.*

4) Considere o conjunto de dados “ans.json” da Agência Nacional de Saúde Suplementar (ANS) disponível em www.lia.ufc.br/~timbo/streaming/ans.json. Esse conjunto de dados contém indicadores relativos à saúde suplementar no Brasil entre os anos de 2000 e 2013, extraídos do portal de dados abertos do governo federal - http://dados.gov.br/dataset/saude-suplementar. As tuplas, por exemplo, possuem a seguinte estrutura: {"valor":10, "municipio_ibge": 171720, "ano": 2013}. Nesse exemplo, temos que 10k reais foram investidos no município de ID 171720, no ano de 2013. Utilizando Apache Spark, leia os dados em questão e elabore scripts Python para responder as seguintes perguntas: (2,0 pontos)

   a) Qual a média de valor investido nos municípios nos anos de 2010 e 2011?
   b) Qual o ID do município do IBGE que recebeu mais aportes, ou seja, mais investimentos ao longo de todos os anos?
   c) Quais os 10 municípios que menos receberam aportes ao longo de todos os anos?

   Obs. O arquivo json está separado por vírgula. Logo, você deve carregá-lo utilizando a opção *.option("multiline", "true")* no método read.

5) A partir do exercício em sala sobre Spark Streaming e utilizando netcat a partir da porta 9999 para produzir os dados, desenvolva um script Python para cada um dos problemas a seguir: (3,0 pontos)

   Sem utilizar janela de tempo:

   a) Exibir apenas as palavras que terminam com um número qualquer. Exemplos: qwe4, des11, cvb0
   b) Exibir a soma dos caracteres não vazios de uma frase inserida via netcat. Exemplo:

      Entrada: Streaming de dados

      Saída: 16 caracteres
   c) Acumular a contagem de palavras à medida que os dados vão chegando via streaming e sumarizar pela quantidade de palavras que existem com um certo número de vogais. Exemplo:

      Entrada: Paraguai, Uruguai, Brasil, Argentina, Cuba, Peru

      Saída:

      2: 3 // (3 palavras com 2 vogais: Brasil, Cuba, Peru)

      4: 1 // (1 palavra com 4 vogais: Argentina)

      5: 2 // (2 palavras com 5 vogais: Paraguai, Uruguai)

   Utilizando janela de tempo:

   d) Exibir apenas nomes de times da série A do brasileiro que chegam via streaming (netcat) em uma janela de 30 segundos
   e) Exibir o número de vezes em que a palavra ‘INFO’ aparece nos últimos 20 segundos
   f) Considerando apenas entradas inteiras via netcat, exibir a média dos números que chegam em uma janela de 15 segundos

6) Utilizando o conceito de janela do Spark Streaming, e o código Kafka disponível em www.lia.ufc.br/~timbo/streaming/producer_consumer.py, exibir o número de tuplas que possuem registros com idade superior a 30 anos, considerando apenas os últimos 30 segundos da janela de tempo. (1,0 ponto)

7) Utilizando os dados de streaming estruturados de atividades físicas vistos em sala, disponível em www.lia.ufc.br/~timbo/streaming/activity-data.zip, crie um script Python para exibir os seguintes campos: User, gt, model, Arrival_Time, filtrados apenas pelo user 'a' e por 'gt = walk' ou 'gt = stand'. Renomei a coluna 'gt' para 'activity' e mostre as informações sendo carregadas via streaming em intervalos de 5 segundos. (0,5 pontos)

8) A partir da integração entre Spark Streaming e KAFKA verificada na disciplina, crie um script Python que atenda aos seguintes requisitos: (2,0 pontos)

   a) Ler os seguintes tópicos: facebook, tweeter, instagram. Caso a mensagem enviada pelo Broker seja do tópico facebook, imprima a mensagem com o <F> no início. Caso seja do tópico tweeter, imprima a mensagem com o <T> no início. Por fim, caso a mensagem seja proveniente do tópico instagram, imprima a mensagem com o <I> no início. Os produtores devem ser serviços criados pelo KAFKA via linha de comando (CLI).

   b) Inserir uma janela de tempo de 20 segundos no exemplo anterior. Dessa forma, apenas mensagens dos últimos 20 segundos serão exibidas.

   c) Criar um quarto tópico denominado other, que irá exibir somente a contagem de palavras que **não** estão na seguinte lista: [streaming, uni7, data, science] nos últimos 20 segundos. Ou seja, caso uma dessas palavras seja enviada via Producer, elas não serão computadas. A exibição deve ser da seguinte forma:

      n palavras válidas nos últimos 20 segundos

Observações:
- Máximo de três pessoas por trabalho.
- A tarefa vale 12pts. Notas superiores a 10 serão arredondadas para o valor máximo: 10.
- Pode-se escolher quaisquer questões para resolução.

Entregáveis:
Para cada questão de implementação:
1. Código da solução
2. Respostas das perguntas
Para questões dissertativas, apenas sua respectiva resposta.

Enviar os entregáveis em formato .zip para o e-mail: timbo.felipe@gmail.com, com o título Streaming de Dados Uni7 - Tarefa Final [nomes-das-pessoas-do-grupo].

Prazo:
09/setembro (quarta-feira) às 23:59h.

Bom trabalho!