

Cronograma [\[editar\]](#)

- [Seminário Curto](#) – 10 minutos
- 24/05/2022
- [Seminário Longo](#) – 50 minutos
- 21/06/2022 - Entrega
- 28/06/2022 - Avaliação dos pares
- **Tutorial**
- 31/05/2022 – 5 páginas
- 07/06/2022 – 10 páginas
- 14/06/2022 – 15 páginas
- [Seminário curto](#) – 10 minutos (warm-up) - Não vale nota, mas é obrigatório. O que é ? Para que serve? Um ou dois exemplos simples.
- [Seminário longo](#) – 50 minutos
 - O que é ? Para que serve? Vários exemplos interessantes.
 - Além de explicar a parte teórica, traga aplicações interessantes. Idealmente, mostre algo funcionando.
 - Em nenhuma hipótese, gaste seu tempo mostrando como instalar a ferramenta. Se a instalação for muito complexa, indique um tutorial.
 - É opcional gravar sua imagem, mas é importante que seu vídeo tenha um áudio razoável com pouco (ou nenhum) ruído.
 - Não deixe para fazer a gravação na última hora. Lembre-se também de assistir o vídeo para verificar se não houveram falhas na gravação.
 - Suas habilidades em edição de vídeo não serão avaliadas, mas podem ser aplicadas. Evite abusos. Lembre-se que o conteúdo é o mais importante.
- Tutorial - Texto
 - PDF – 15 páginas
 - Formato SBC [\[link\]](#)
 - Explicação didática sobre o tópico.
 - MUITOS exemplos interessantes.
 - Cuidado com plágio.
 - Entregas parciais obrigatórias 5, 10 e 15 páginas.

[SLIDE 1] Olá, eu sou o felipe, eu sou o breno e eu sou o joao. Somos alunos do curso de ciencia da computação da utfpr. Em relação a diciplina de banco de dados 2, nosso projeto é sobre o O Apache Kafka. Mas "o que é isso? para que que ele serve? quando eu vou utilizar? e aonde eu vou utilizar?" Acreditamos que essas são as principais dúvidas para quem que nunca ouviu falar nessa tecnologia que de fato é fantástica. Entao o objetivo dessa apresentação é esclarecer pra todos vocês essas questões.

[SLIDE 2 - Breno] Vamos lá, primeiramente a gente tem que entender que nós temos diversos problemas para tratar no mundo moderno. Nesse problemas nós temos os dados espalhados e sendo enviados de lá para cá entre sistemas. Fundamentalmente vamos

imaginar que eu tenho um sistema "A" e eu tenho um sistema "B". Esse sistema "A" quer mandar uma mensagem para o sistema "B", então o que normalmente a gente imagina é que esse cara vai falar direto com outro. Porém o que aconteceria se o nosso sistema "B" estiver fora do ar é que essa mensagem provavelmente vai ser perdida, daí para a gente garantir que essa mensagem não vai ser perdida, a gente pode colocar essa mensagem em um "broker".

"Broker" é um sujeito que ele é o intermediário da comunicação entre esses dois sistemas, ou seja, eu joga essa mensagem para esse broker e quando o outro sistema que quer receber essa mensagem estiver no ar, ele pode consumir essa mensagem. Então uma das principais utilidades do apache kafka é ser esse broker. O Apache kafka tem um conceito que ele chama de "tópicos", e o que esse tópico faz basicamente é: eu tenho um sujeito que é um produtor de conteúdo, e eu tenho um sujeito que é o consumidor desse conteúdo. O produtor joga essa mensagem em um tópico e essa mensagem, ela fica lá guardada até que um consumidor que vai ler esse tópico consuma essas mensagens.

O conceito do kafka ele é muito parecido para quem olha assim pela primeira vez como um sistema o tradicional de filas, porém ele tem algumas peculiaridades que fazem total diferença para você escolher ele no dia a dia para trabalhar com seus sistemas. A primeira coisa que você tem que entender é que quando um consumidor lê essa mensagem, essa mensagem, ao invés dela ser jogada fora ela fica armazenada em um banco de dados ali do apache kafka. Dessa forma, se em algum momento eu precisar reprocessar ou consumir novamente essa mensagem, eu consigo também consumir ela.

Fazer analogia com buffers assíncronos em problemas de produtor e consumidor

Acredito que essa seria a melhor analogia possível.

[SLIDE 3 - João] Uma outra coisa muito interessante em relação ao apache kafka é que ele funciona de forma distribuída ou seja ele tem uma resiliência muito grande porque cada vez que você praticamente cria um tópico você vai falar assim pro apache kafka: "olha meu amigo eu vou criar um tópico", que é o lugar onde a gente joga as nossas mensagens e esse tópico ele vai ter três partições, ou seja é como se ele divide se esse tópico em três partes, em três gavetinhas para você colocar as mensagens. Mas essas gavetinhas normalmente elas ficam espalhadas em outras máquinas, ou seja se uma máquina cair, por exemplo, eu garanto que as outras gavetinhas estão no ar para que as mensagens ainda estejam livres. Mas daí você deve estar pensando: "Mas aquele camarada que caiu, eu vou perder aquelas mensagens?" A resposta é: Não, porque o kafka ele tem um camarada que a gente chama de "**Replication factor**".

O replication factor é tipo assim, você fala o seguinte: "Olha meu amigo, eu tenho três partições, ou seja, eu vou falar que o meu replication factor é de três", então isso

significa que, apesar de eu ter as minhas partições, que é onde as gavetinhas onde cada mensagem fica guardada, eu vou sempre guardar uma cópia dessa partição um outras máquinas que estão distribuídas. Dessa forma, se em algum momento uma máquina ela cair, outra máquina que tem aquela mesma partição que é aquela cópia vai sumir e nenhuma mensagem vai ser perdida.

[SLIDE 4 - Felipe] Além disso, o kafka que ele tem recursos como kafka-connect que ele utiliza conectores que simplesmente você pluga ali e ele começa consumir mensagens de fontes externas. Um exemplo disso seria tipo: Eu tenho twitter e eu quero pegar ali uma certa hashtag. Ou seja, eu configuro o meu kafka-connect e ele vai ficar lendo o twitter o tempo inteiro e toda vez que ele achar a hashtag que a gente escolher, ele vai jogar no kafka dentro do tópico que a gente escolher.

Outro exemplo seria em relação aos banco de dados como o por exemplo o mongoDB ou mysql em que, a cada alteração em seu banco, o kafka connect que estará no processo irá transmitir essa alteração para todos que estiverem ouvindo.

Com isso, vemos que o apache kafka é realmente um sistema fantástico para você conseguir trabalhar com ferramentas de mensageria, inclusive trabalhar com Pub/Sub (Publish-subscribe), ou seja, todo mundo fica lendo um canal e um produtor produz ali naquele canal, ou seja, nesse tópico e todo mundo consegue ler essas mensagens. Dessa forma, se você está pensando em alguma solução para você fazer o seu sistema se comunicar de forma segura e eficiente e que você ainda, se precisar, consiga reprocessar essas mensagens, o apache kafka é uma ótima solução para você considerar.

[SLIDE 5] Maria deseja presentear seu amigo André com duas laranjinhas, mas André dorme frequentemente e por vezes, durante longos períodos, então, para ter certeza de que André receberá seu presente ao acordar, José distribui 2 laranjas entre as mesas e cestas de fruta na casa de André, e para caso alguma laranja apodreça enquanto ele dorme, coloca mais 2 como precaução.

Temos em vista que as laranjas são os dados, que André é o consumidor e José o produtor, as cestas as partições, as mesas são os brokers, e a casa o tópico (presentear).

Entao esse é o nosso trabalho. Espero que tenhamos explicado de forma clara e objetiva, alguns dos conceitos básicos do apache kafka.