

## Trabalho Prático 1 (TP 01) – Disciplina de Sistemas Distribuídos

Neste trabalho, o aluno utilizará o conceito de middlewares (<https://en.wikipedia.org/wiki/Middleware>) para construir sua 1ª versão de uma solução distribuída. O aluno poderá optar por apenas executar um algoritmo acordado previamente com o professor ou implementar um algoritmo próprio na solução de middleware sendo utilizada ou implementar o algoritmo de ordenação ilustrado em sala de aula.

Nota máxima 8.0 para o aluno que apenas executar um algoritmo acordado previamente com o professor.

Nota máxima 9.0 para o aluno que buscar programar algo no middleware escolhido. Favor combinar com o professor o algoritmo escolhido para ver grau de complexidade. Ex. Não vale fazer apenas uma sentença SQL no Cassandra (<https://cassandra.apache.org/>) do tipo “SELECT ALL FROM TABLE T”.

Também será avaliado a instalação do middleware em nuvens públicas (Google cloud, AWS etc.), ou seja, caberá ao aluno instalar todos os pacotes e dependências da solução escolhida num cluster de máquinas Linux em alguma infraestrutura pelo mundo. Instalar recursos avançados, tal como tolerância a falhas, será muito valorizado na nota final do TP 01.

Nota máxima 10 pontos para o aluno que conseguir, além de instalar o middleware no cluster com tolerância a falhas e outras ‘features’, conseguir instalar um cluster de VMs e nestas um conjunto de contêineres Docker (<https://www.docker.com/resources/what-container>) e orquestradores, tais como Kubernetes (<https://kubernetes.io/pt/>) ou Docker Swarm (<https://docs.docker.com/engine/swarm/>).

As soluções de middleware sugeridas:

1. Apache Spark SQL - (<https://spark.apache.org/sql/>)
2. Apache Spark GraphX (<https://spark.apache.org/graphx/>)
3. Apache Spark MLlib (<https://spark.apache.org/mllib/>)
4. Apache Spark Stream (<https://spark.apache.org/streaming/>)  
Estes primeiros 4 grupos (até 12 alunos) podem se juntar para colocar o Spark na nuvem com tolerância a falhas e rodando no Docker, por exemplo. Podem atingir nota 10 trabalhando juntos!
5. Apache Kafka (<https://kafka.apache.org/>)
6. RabbitMQ (<https://www.rabbitmq.com/>)
7. Redis Pub/Sub (<https://redis.io/topics/pubsub>)
8. MongoDB notification usando realm (<https://docs.mongodb.com/realm/tutorial>) ou usando stream (<https://docs.mongodb.com/manual/changeStreams/>)
9. Apache Cassandra (<http://cassandra.apache.org/>)
10. Apache Ignite (<https://ignite.apache.org/>)
11. Apache Solr (<https://solr.apache.org/>)

12. Apache Syncope (<http://syncope.apache.org/>)
13. Apache Flink (<https://flink.apache.org/>)
14. Apache Druid (<https://druid.apache.org/druid>)
15. Apache Zookeeper (<https://zookeeper.apache.org/>)

**Entrega:** Faremos um workshop de BCC 362 para que cada equipe com 3 alunos apresente nifiseu TP 01 aos demais colegas e ao professor. Almejamos ampla participação de TODOS no workshop para aprendermos um pouco das soluções de middleware para computação em larga escala que estão sendo usadas na atualidade.

**USAR CLOUDS PÚBLICAS PELO MUNDO. USAR MAIS DE 5 MÁQUINAS. SEJA QUAL FOR O ALGORITMOS USADO NO TESTE DO MIDDLEWARE, NÃO USAR INPUTS PEQUENOS. SE PLANEJE PARA NÃO FICAR SEM TEMPO DE APRENDER A MEXER COM PLATAFORMAS EM NUVEM, O QUE IMPLICA EM FICAR SEM TESTAR SUA SOLUÇÃO!!**

A apresentação do TP é fundamental, portanto atenção para sua confecção.

Neste TP 01 devemos:

- 1) Descrever a solução de middleware utilizada (para que serve, para que não serve, onde é mais utilizada no mercado, quais os tipos de aplicações podemos construir com tal middleware, qual a maior aplicação usando tal middleware e quantas máquinas há em tal cluster e outras curiosidades....) (em torno de 5 minutos de apresentação)
- 2) Apresentar como a solução de middleware funciona internamente, ou seja, como sua arquitetura está organizada em termos de componentes, hierarquias dos componentes, camadas existentes e serviços prestados em cada camada. (em torno de 5 minutos de apresentação)
- 3) Descrever as formas de instalação existentes, ou seja, como instalar o middleware em apenas uma máquina (se existir), como o instalar em um cluster, como o instalar numa nuvem pública da Amazon (AWS) ou Microsoft (Azure), como ativar serviços avançados, tal como tolerância a falhas e outros. Conseguiu instalar o cluster usando contêineres? E orquestradores, entendeu e conseguiu usar? (em torno de 5 minutos de apresentação)
- 4) Apresentar a aplicação escolhida para o TP 01 (conceitos, origem, aplicabilidade, desafios da versão distribuída, etc.), apresentar a aplicação rodando usando o middleware e sob um cluster. Como a aplicação escala com INPUTS maiores? É importante ressaltar o que ocorre se uma máquina cai abruptamente ou caso seja necessário aumentar o cluster sem interromper o serviço prestado ou qualquer outra demanda mais avançada. (em torno de 5 minutos de apresentação)
- 5) Responder aos questionários. (em torno de 10 minutos de apresentação)