

# Lista 4 -Arquitetura de Software

**Aluno:** Lucas Lage e Silva

**Data:** 20/05/2023

1) A atomicidade é uma característica que pode ser garantida ao se aplicar uma arquitetura monolítica visto que esse padrão arquitetural garante alguns fatores que contribuem para atomicidade como o bloqueio de recursos e a utilização de bancos de dados. Primeiramente, o bloqueio de recursos consiste na capacidade do sistema que, ao executar sua totalidade como um único processo, garante que não haverá acesso simultâneo de recurso, ou seja, o sistema não irá acessar de maneira concorrente aliado ou coisa do tipo, o que garantiria a atomicidade das operações do sistema. Em segundo lugar, a utilização de transações de bancos de dados na aplicação também consiste de uma técnica para aumentar a atomicidade das operações do sistema pois gera uma unificação da persistência dos dados e portanto garante que não haja replicação ou inconsistências nas informações.

2) Por mais que o protocolo 2PC seja uma maneira mais adequada para garantir maior atomicidade em sistemas distribuídos, ainda existem alguns desafios atrelados a implementação desse modelo para que a atomicidade realmente tenha um ganho considerável. Em primeiro lugar, é necessário que se estabeleça um coordenador centralizado, o qual fica responsável por lidar com todo o processo transacional entre os sistemas, o que requer desse coordenador uma disponibilidade grande visto que caso haja algum problema nele, isso pode acarretar em instabilidades entre todos os sistemas. Em segundo lugar, a rede pode desempenhar um importante fator para garantia da atomicidade, visto que esse protocolo parte do pressuposto que existe uma comunicação confiável entre os sistemas e o coordenador, de maneira que instabilidades podem fazer com que transações sejam canceladas, ou até que se percam e que, dessa maneira, informações e processos entre os sistemas não sejam devidamente gerenciados como deveriam. Em terceiro lugar, a utilização desse protocolo deve ser feita tomando os devidos cuidados visto que, pela sua implementação, ele pode gerar o bloqueio de recurso além de prejudicar a escalabilidade e até a disponibilidade de um serviço distribuído visto que a execução completa da funcionalidade utilizada depende da ocorrência completa do processo do coordenador com os sistemas.

3) Ela não seria considerada atômica visto que o fator que gera a atomicidade das operações está na confiabilidade gerada a partir da 'confirmação' que é garantida entre as etapas da Saga, sendo que o mecanismo de compensação é um forte indicador de que houve algum problema e que, portanto, é necessário retroceder e realizar novamente a operação pois possível ela não foi atômica.

4) A vantagem da utilização do padrão saga para transações de longa duração está em sua capacidade de, ao dividir a transação em etapas com compensações associadas, a Saga permite que as transações de longa duração sejam tratadas de forma mais granular e flexível. As operações podem ser executadas em momentos diferentes, permitindo que o sistema se recupere de erros e mantenha a atomicidade mesmo em cenários complexos e prolongados.