
IDEMPOTÊNCIA COMO ESTRATÉGIA DE CONFIABILIDADE EM SISTEMAS DISTRIBUÍDOS

PEREIRA, André Luís Santos¹  , ALMEIDA, Felipe Queiroz de¹  

¹Departamento de Computação, Centro Universitário Fieo, Osasco, São Paulo, Brasil.

*e-mail de correspondência: 98880@unifio.br - André Luís Santos Pereira

Recebido: 05/02/2026 - Aceito: /03/2026 - Publicado online: 23/03/2026

Resumo

Falhas como *timeouts*, reenvio automático de requisições e reprocessamento de mensagens fazem parte do comportamento normal da infraestrutura de sistemas distribuídos modernos. Nesse contexto, operações idempotentes tornam-se essenciais para garantir consistência e previsibilidade no estado do sistema. Idempotência refere-se à propriedade segundo a qual múltiplas execuções de uma mesma operação produzem o mesmo resultado observável. Este estudo apresenta uma análise prática da idempotência em sistemas distribuídos, com foco em estratégias arquiteturais para o tratamento de operações duplicadas e seus impactos na confiabilidade do sistema. São exploradas abordagens como chaves de idempotência, controle baseado em estado e identificação única de eventos. A análise indica que a adoção consciente da idempotência como decisão arquitetural contribui significativamente para a confiabilidade, resiliência e recuperação segura de falhas em arquiteturas distribuídas.

Palavras-chave: Sistemas distribuídos; Idempotência; Arquitetura de *software*; Confiabilidade; Microsserviços

Abstract

Failures such as timeouts, automatic request retries, and message reprocessing are common behaviors in modern distributed systems infrastructure. In this context, idempotent operations become essential to ensure system consistency and predictability. Idempotence refers to the property in which multiple executions of the same operation produce the same observable result. This study presents a practical analysis of idempotency in distributed systems, focusing on architectural strategies for handling duplicate operations and their impact on system reliability. Approaches such as idempotency keys, state-based control, and unique event identification are explored. The analysis indicates that incorporating idempotency as an architectural decision significantly improves reliability, resilience, and safe fault recovery in distributed architectures.

Keywords: *Distributed systems; Idempotence; Software architecture; Reliability; Microservices.*

INTRODUÇÃO

Sistemas distribuídos operam em contextos nos quais a ocorrência de falhas parciais é inerente (1). Interrupções de conectividade, timeouts de requisição, duplicação de mensagens e mecanismos automáticos de retentativa (*retries*) constituem comportamentos recorrentes na infraestrutura contemporânea (1,4). Esses eventos frequentemente fazem com que operações sejam executadas múltiplas vezes, o que pode levar a estados inconsistentes se os sistemas não forem projetados para lidar com tais cenários.

Em sistemas monolíticos tradicionais, é comum assumir que as operações são executadas exatamente uma única vez (2). No entanto, essa premissa raramente se sustenta em ambientes distribuídos (1), onde os serviços se comunicam por meio de redes e sistemas de mensageria assíncrona, sujeitos a falhas e incertezas inerentes.

A idempotência surge como uma resposta a esse desafio ao assegurar que múltiplas execuções de uma mesma operação resultem em um estado final consistente (3). Quando corretamente implementadas, operações idempotentes permitem que os sistemas tolerem retentativas, mensagens duplicadas e instabilidades de rede sem comprometer a integridade dos dados.

Este artigo discute o papel da idempotência como uma estratégia arquitetural em sistemas distribuídos. A discussão foca em cenários comuns de falha, estratégias práticas de implementação e considerações arquiteturais para a aplicação da idempotência em camadas de serviço e interfaces de sistema.

Como contribuição, este estudo apresenta uma sistematização prática das principais estratégias de implementação de idempotência em sistemas distribuídos, discutindo suas aplicações em diferentes camadas arquiteturais e destacando seus impactos na confiabilidade e resiliência de sistemas modernos. Além disso, propõe uma análise comparativa entre abordagens, contribuindo para a tomada de decisão arquitetural em contextos distribuídos.

MATERIAIS E MÉTODOS

Este estudo caracteriza-se como uma revisão narrativa da literatura, com foco em práticas arquiteturais relacionadas à idempotência em sistemas distribuídos. A seleção das fontes foi realizada com base em obras clássicas e amplamente referenciadas na área de sistemas distribuídos e engenharia de software, incluindo livros e publicações técnicas reconhecidas.

Foram considerados trabalhos que abordam padrões de arquitetura, confiabilidade e comunicação distribuída, com ênfase em estratégias para tratamento de falhas e duplicidade de operações. A análise foi conduzida de forma qualitativa, buscando identificar padrões recorrentes, vantagens, limitações e contextos de aplicação das principais abordagens de idempotência.

Exemplos conceituais foram utilizados para ilustrar cenários típicos de duplicação de operações e os mecanismos arquiteturais empregados para garantir consistência e confiabilidade.

RESULTADOS E DISCUSSÃO

A análise das estratégias de idempotência demonstra que sua aplicação contribui significativamente para a resiliência de sistemas distribuídos. Técnicas como chaves de idempotência permitem identificar requisições repetidas e evitar execução redundante de operações (2).

Embora as estratégias de idempotência apresentem benefícios claros em termos de confiabilidade, sua implementação não é isenta de desafios. O uso de chaves de idempotência, por exemplo, pode introduzir sobrecarga de armazenamento e complexidade na gestão de estado, especialmente em sistemas de alta escala.

De forma comparativa, observa-se que chaves de idempotência são mais adequadas para operações síncronas e APIs, enquanto o controle baseado em estado é mais eficaz em cenários de domínio bem definido. Já a identificação única de eventos se destaca em arquiteturas orientadas a eventos, onde a duplicidade de mensagens é um comportamento esperado da infraestrutura.

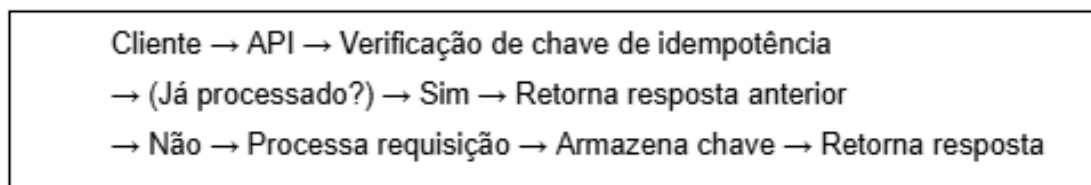
O controle baseado em estado, embora eficaz para garantir a consistência de transições de domínio (2), depende de modelagens cuidadosas e pode não ser suficiente em cenários altamente distribuídos e assíncronos. De forma complementar, a identificação única de eventos é amplamente adotada em arquiteturas orientadas a eventos para evitar o reprocessamento de mensagens (4), porém exige mecanismos adicionais para rastreamento e persistência de eventos processados.

Dessa forma, a escolha da estratégia mais adequada depende do contexto arquitetural, dos requisitos de consistência e dos níveis aceitáveis de complexidade e custo operacional. Essa análise evidencia que a idempotência deve ser tratada como uma decisão arquitetural fundamental, e não como uma solução universal.

Adicionalmente, no contexto arquitetural, observou-se que a aplicação da idempotência em diferentes camadas arquiteturais — especialmente nas camadas de serviço e adaptadores — permite equilibrar controle de domínio e filtragem de duplicidades provenientes da infraestrutura (2,5). Esses resultados reforçam que a idempotência deve ser tratada como uma decisão arquitetural fundamental em sistemas distribuídos modernos.

O fluxo consolidado de processamento idempotente é apresentado na Figura 1.

Figura 1 – Fluxo de processamento idempotente em requisições distribuídas



Fonte: Autoria própria, 2026.

Limitações do Estudo

Este estudo apresenta limitações por se tratar de uma revisão narrativa, sem validação empírica por meio de experimentos ou estudos de caso. Além disso, a análise baseia-se em fontes selecionadas da literatura e na interpretação conceitual

das estratégias, o que pode não contemplar todas as variações existentes em ambientes reais.

Trabalhos futuros podem explorar validações experimentais e estudos de caso para aprofundar a análise das abordagens discutidas.

CONCLUSÃO

Falhas e retentativas são características intrínsecas aos sistemas distribuídos. Projetar sistemas sob a premissa de execução única mostra-se arriscado e frequentemente incompatível com a realidade desses ambientes. A idempotência fornece um mecanismo prático e eficaz para garantir consistência em cenários de execução duplicada. Ao incorporá-la como decisão arquitetural, é possível reduzir o impacto de falhas, permitir retentativas seguras e aumentar a confiabilidade geral das aplicações distribuídas. Este estudo reforça a importância da idempotência como um elemento central no design de sistemas distribuídos resilientes.

DECLARAÇÃO DE DISPONIBILIDADE DE DADOS

Os dados utilizados neste estudo estão integralmente disponíveis no corpo do artigo.

REFERÊNCIAS

1. Kleppmann M. **Designing data-intensive applications**. Sebastopol: O'Reilly Media; 2017.
2. Newman S. **Building microservices**. 2nd ed. Sebastopol: O'Reilly Media; 2021.
3. Helland P. **Idempotence is not a medical condition**. ACM Queue. 2007;5(12):18-22.
4. Hohpe G, Woolf B. **Enterprise integration patterns: designing, building, and deploying messaging solutions**. Boston: Addison-Wesley; 2004.

5. Cloud Native Computing Foundation. **Top 5 resiliency trends of 2023** [Internet]. 2023 [cited 2026 Mar 15]. Disponível em: <<https://www.cncf.io/blog/2023/09/22/top-5-resiliency-trends-of-2023/>>. Acesso em 17/10/2025.