INTERFACE TECNOLÓGICA

COMPARAÇÃO ENTRE ARQUITETURAS MONOLÍTICAS E MICROSSERVIÇO: vantagens e desafios.

COMPARISON BETWEEN MONOLITHIC AND MICROSERVICES ARCHITECTURES: advantages and challenges.

Vinicius de Souza Borges–viniborges2005@gmail.com
Fatec Taquaritinga – Taquaritinga – São Paulo – Brasil

Guiliano Scombatti Pinto – giuliano.pinto@fatec.sp.gov.br
Fatec Taquaritinga – Taquaritinga – São Paulo – Brasil

DOI: 10.31510/infa.v22i2.2379
Data de submissão: 26/09/2025
Data do aceite: 02/12/2025
Data da publicação: 20/12/2025

RESUMO

Este estudo visa conduzir uma análise comparativa entre as arquiteturas de software monolítica e de microsserviços, enfatizando suas características, benefícios e desafios mais relevantes. O propósito é ajudar profissionais e estudantes a escolher a abordagem mais apropriada para diversos contextos de aplicação. O estudo fundamenta-se em uma revisão da literatura de livros e artigos científicos, utilizando critérios como acoplamento, coesão, escalabilidade, desempenho, manutenção e complexidade. Os resultados mostram que, apesar de a arquitetura monolítica ser mais simples de implementar, ter um custo inicial menor e ser fácil de gerenciar em sistemas pequenos, ela apresenta limitações importantes em ambientes que exigem alta escalabilidade e resiliência. Por outro lado, a arquitetura de microsserviços oferece maior flexibilidade, escalabilidade granular e isolamento de falhas, tornando-se apropriada para sistemas complexos e em constante evolução. No entanto, requer uma infraestrutura sólida, maior maturidade técnica e práticas avançadas de monitoramento e orquestração. Conclui-se que não existe uma solução universalmente superior, mas sim modelos que se adaptam melhor às demandas do projeto, à experiência da equipe e aos recursos disponíveis. Portanto, a seleção arquitetural deve ser baseada em critérios técnicos e organizacionais, levando em conta os *trade-offs* de cada estratégia para garantir sustentabilidade, eficiência e adaptabilidade durante todo o ciclo de vida do sistema.

Palavras-chave: Arquitetura monolítica. Microsserviços. Escalabilidade. Desempenho. Complexidade.

ABSTRACT

This study aims to provide a comparative analysis between monolithic and microservices software architectures, highlighting their main characteristics, advantages, and challenges, in

order to support professionals and students in choosing the most suitable approach for different application contexts. The research is based on a literature review of books and scientific articles, using criteria such as coupling, cohesion, scalability, performance, maintenance, and complexity. The results indicate that the monolithic architecture, although simpler to implement, with lower initial cost and easier management in small systems, presents significant limitations in environments that require high scalability and resilience. On the other hand, the microservices architecture demonstrates greater flexibility, granular scalability, and fault isolation, making it more suitable for complex and constantly evolving systems, but it demands robust infrastructure, higher technical maturity, and advanced monitoring and orchestration practices. It is concluded that there is no universally superior solution, but rather models that best fit according to the project's requirements, the team's experience, and the available resources. Therefore, the architectural choice should be guided by technical and organizational criteria, considering the trade-offs of each approach to ensure sustainability, efficiency, and adaptability throughout the system's lifecycle.

Keywords: Monolithic architecture. Microservices. Scalability. Performance. Complexity.

1 INTRODUÇÃO

A criação de um sistema demanda a escolha de arquitetura que impacta diretamente na sua escalabilidade, manutenção, desempenho e clico de vida. Neste cenário, duas abordagens se destacam: a arquitetura monolítica e a arquitetura de microsserviços. A primeira, presente em diversos projetos, concentra todas as funcionalidades em um único sistema. Já a segunda, mais recente, ajustada as necessidades de flexibilidade, separa a aplicação em serviços independentes que comunicam entre si (Fowler, 2015; Newman, 2015).

A questão que fundamenta esta pesquisa é: quais são as vantagens e desvantagens da adoção das arquiteturas monolítica e de microsserviços em relação a diferentes contextos de aplicação? Com a crescente necessidade de soluções que sejam escaláveis e adaptáveis, vale a pena explorar em que contextos cada uma das abordagens se revela mais eficaz e quais são as consequências práticas da escolha arquitetural (Silva, 2020; Görgülü, 2022).

Este trabalho tem como objetivo geral realizar uma comparação entre as arquiteturas monolítica e de microsserviços, explorando suas características, vantagens e desvantagens, para que tanto profissionais quanto estudantes possam tomar decisões mais embasadas em relação à melhor solução a ser adotada. Os objetivos específicos incluem: (i) identificar os pontos fortes

e fracos de cada abordagem; (ii) analisar situações práticas onde cada arquitetura é mais adequada; (iii) comparar aspectos como escalabilidade, facilidade de manutenção, desempenho e complexidade; e (iv) fornecer uma análise crítica que ajude na decisão em projetos de software (Rozanski; Woods, 2025).

A justificativa para o estudo baseia-se na necessidade de aprofundar o entendimento sobre as consequências das escolhas arquiteturais em sistemas que estão se tornando cada vez mais complexos. A decisão de permanecer com um sistema monolítico ou migrar para microsserviços traz riscos e oportunidades que podem afetar diretamente a eficiência do desenvolvimento, a adaptabilidade das equipes e a durabilidade da aplicação a longo prazo. Dessa forma, este estudo visa contribuir para uma perspectiva mais crítica e fundamentada sobre o assunto.

2 FUNDAMENTAÇÃO TEÓRICA

Para compreender a organização e a manutenção dos sistemas durante seu ciclo de vida, é essencial analisar as arquiteturas de software. Cada abordagem arquitetural possui características distintas que impactam diretamente a escalabilidade, o desempenho, a manutenção e a flexibilidade das aplicações. A arquitetura monolítica e a arquitetura de microsserviços são duas das principais abordagens, cada uma oferecendo uma perspectiva distinta sobre a organização de sistemas de software.

2.1 Arquitetura Monolítica

A arquitetura monolítica é definida pela criação de um sistema como uma única unidade coesa, na qual todos os módulos e funcionalidades estão incorporados no mesmo código-base e utilizam os mesmos recursos de execução (Silva, 2020). Nesse modelo, as camadas de apresentação, lógica de negócios e acesso a dados estão fortemente interligadas, fazendo com que a aplicação dependa de uma estrutura centralizada.

Historicamente, esse tipo de arquitetura tem sido bastante utilizada em projetos de menor porte devido à sua simplicidade de desenvolvimento, implantação e manutenção (Rozanski; Woods, 2025). A uniformidade de sua estrutura possibilita que os desenvolvedores entendam de forma rápida o funcionamento do sistema, além de simplificar processos como testes e gerenciamento de versões.

No entanto, essa metodologia possui limitações consideráveis em contextos mais complexos. A escalabilidade individual de partes do sistema é dificultada pelo forte acoplamento entre os componentes, o que também aumenta o risco de falhas em cascata. Ademais, a recompilação e redistribuição de todo o sistema são necessárias para qualquer atualização ou alteração, o que afeta diretamente a agilidade do ciclo de desenvolvimento. (Görgülü, 2022).

Logo, a arquitetura monolítica pode ser apropriada para sistemas pequenos e estáveis, mas sua utilização pode se transformar em um entrave em projetos que exigem alta disponibilidade, escalabilidade e evolução constante.

2.2 Arquitetura de Microsserviços

A arquitetura de microsserviços se apresenta como uma opção ao modelo monolítico, sugerindo a divisão do sistema em serviços independentes, pequenos e autônomos, que se comunicam por meio de interfaces claramente definidas, normalmente APIs (Fowler, 2015). Cada microsserviço tem a responsabilidade de uma funcionalidade específica do sistema, o que permite que seja desenvolvido, implantado e escalado de maneira independente (Newman, 2015).

Essa estratégia favorece um baixo acoplamento e uma alta coesão, permitindo maior flexibilidade no processo de desenvolvimento e manutenção. Desde que cumpram os protocolos de comunicação definidos, equipes distintas podem atuar simultaneamente em serviços diferentes, empregando inclusive linguagens e tecnologias diversas. Ademais, a escalabilidade se torna mais granular, possibilitando a expansão apenas dos serviços essenciais de acordo com a demanda (Görgülü, 2022).

Contudo, a arquitetura de microsserviços também traz dificuldades. A fragmentação do sistema resulta em um aumento da complexidade na gestão, demandando mecanismos sofisticados para monitoramento, orquestração e comunicação entre os serviços. Nesse cenário, aspectos como latência, tolerância a falhas e segurança tornam-se mais importantes (Newman, 2015).

Embora haja desafios, os microsserviços vêm se firmando como uma estratégia bastante empregada em sistemas modernos, especialmente em aplicações que demandam escalabilidade, resiliência e evolução constante. Essa arquitetura proporciona maior adaptabilidade às mudanças frequentes do mercado e às demandas dos usuários ao distribuir responsabilidades.

3 PROCEDIMENTOS METODOLÓGICOS

A metodologia utilizada no presente artigo foi a revisão bibliográfica. Os materiais considerados foram livros, artigos, teses, dissertações e publicações eletrônicas

Inicialmente, foi realizado um levantamento para identificar as principais obras relacionadas ao tema. O material foi identificado por meio de buscas realizadas em bases acadêmicas a partir do Scholar, ACM Digital Library e publicações de autores reconhecidos na área. Posteriormente, realizou-se a leitura dos resumos dos materiais para identificar os mais adequados ao propósito do presente artigo.

Em seguida, foi feita a leitura dos matérias selecionados, buscando identificar contribuições teóricas e práticas que sustentassem a análise comparativa entre as arquiteturas monolítica e de microsserviços.

Por fim, os resultados e discussões foram sistematizados e são apresentados que são apresentados na seção a seguir.

4 RESULTADOS E DISCUSSÃO

4.1 Comparação entre as arquiteturas

A análise comparativa entre arquiteturas monolíticas e de microsserviços permite entender como suas particularidades afetam o desenvolvimento, a manutenção e a evolução dos sistemas. A seguir, são abordados os critérios de comparação mais relevantes

4.1.1 Acoplamento e Coesão

Na arquitetura monolítica, os módulos são fortemente acoplados, facilitando a integração inicial, mas dificultando mudanças estruturais, pois qualquer alteração pode afetar toda a aplicação. Já nos microsserviços, busca-se baixo acoplamento e alta coesão: cada serviço possui uma responsabilidade específica e pode ser atualizado de forma independente, favorecendo a evolução contínua do sistema.

4.1.2 Escalabilidade

Monólitos exigem a replicação completa da aplicação para escalar, o que pode gerar desperdício de recursos. Microsserviços permitem escalar apenas os serviços mais demandados, oferecendo escalabilidade granular e maior eficiência no uso da infraestrutura.

4.1.3 Implementação e ciclo de desenvolvimento

Arquiteturas monolíticas têm implantação inicial mais simples, mas atualizações tornam-se mais lentas conforme o sistema cresce, pois exigem redistribuir toda a aplicação. Nos microsserviços, equipes podem desenvolver e implantar serviços paralelamente, acelerando o ciclo de entrega, embora demandem pipelines de CI/CD mais robustos.

4.1.4 Tolerância a falhas e resiliência

Em monólitos, falhas em um módulo podem comprometer todo o sistema devido à ausência de isolamento. Microsserviços oferecem maior resiliência, já que serviços independentes podem continuar operando mesmo se outro falhar, desde que existem mecanismos adequados de fallback e monitoramento.

4.1.5 Complexidade de gerenciamento

Monólitos são simples de operar, mas tornam-se difíceis de manter à medida que crescem. Microsserviços, por sua vez, aumentam a complexidade operacional, exigindo ferramentas de orquestração, monitoramento e controle de versões, além de maior maturidade técnica da equipe.

4.1.6 Performance e consumo de recursos

Monólitos tendem a ter melhor desempenho interno pela comunicação direta entre módulos. Microsserviços permitem melhor distribuição de carga, mas sofrem com latência de rede e maior consumo de recursos, dependendo fortemente da qualidade da infraestrutura.

4.2 Vantagens de cada abordagem

Cada arquitetura de software oferece vantagens que podem justificar sua utilização em variados contextos. A decisão entre a estratégia monolítica e a de microsserviços deve levar em conta as particularidades do projeto, a equipe participante e as necessidades de escalabilidade e manutenção.

4.2.1 Vantagens da arquitetura monolítica

A arquitetura monolítica é conhecida por sua facilidade de implementação e gerenciamento inicial. Como todo o sistema é desenvolvido a partir de uma única base de código, o processo de desenvolvimento tende a ser mais simples, demandando menos esforço em termos de configuração e integração (Silva, 2020).

Outras vantagens incluem:

- **Implantação simplificada:** somente um pacote de software precisa ser distribuído, diminuindo as complexidades de orquestração (Rozanski; Woods, 2025);
- **Desempenho interno eficaz:** a interação entre módulos acontece na memória, com latência reduzida;
- **Custo inicial de infraestrutura reduzido:** não há necessidade de vários servidores ou de ferramentas sofisticadas de monitoramento e integração;
- **Curva de aprendizado reduzida:** equipes com menos experiência conseguem entender e operar o sistema com mais facilidade.

Dessa forma, a arquitetura monolítica é recomendada para sistemas de pequeno a médio porte, estáveis e que não exigem uma escalabilidade considerável.

4.2.2 Vantagens da arquitetura de microsserviços

A arquitetura de microsserviços proporciona mais flexibilidade e escalabilidade, sendo particularmente benéfica para sistemas complexos e de grande escala (Newman, 2015).

Entre seus principais benefícios, estão:

- **Escalabilidade granular**: somente os serviços mais requisitados podem ser duplicados, otimizando recursos (Görgülü, 2022);
- **Isolamento de falhas:** dificuldades em um serviço não afetam toda a aplicação (Fowler, 2015);
- **Agilidade no desenvolvimento:** as equipes podem atuar simultaneamente em diversos serviços, diminuindo o prazo de entrega;
- **Flexibilidade tecnológica:** é possível desenvolver cada serviço em diferentes linguagens e frameworks, desde que as interfaces de comunicação sejam respeitadas;
- **Evolução contínua:** a descentralização possibilita a atualização e troca de serviços sem causar grandes efeitos no sistema global.

Assim, os microsserviços são apropriados para aplicações modernas que demandam resiliência, escalabilidade e habilidade para se adaptar rapidamente às transformações do mercado.

Tabela 1: Comparação das vantagens entre arquiteturas monolítica e de microsserviços

Tipo	Monolítica	Microsserviços
Facilidade inicial	Simples de implementar	Maior complexidade inicial
Escalabilidade	Limitada	Granular
Resiliência	Baixa	Alta
Custo inicial	Menor	Maior
Evolução contínua	Difícil	Flexível

Fonte: Elaborada pelo autor

4.3 Desafios e limitações

Apesar de suas vantagens, tanto a arquitetura monolítica quanto a de microsserviços apresentam limitações que devem ser cuidadosamente avaliadas no momento da escolha.

4.3.1 Desafios da arquitetura monolítica

O forte acoplamento entre módulos é a principal desvantagem da arquitetura monolítica, o que afeta a escalabilidade e a flexibilidade. A manutenção de sistemas grandes se torna complexa e arriscada, pois qualquer mudança pode afetar todo o sistema (Silva, 2020).

Outros obstáculos incluem:

- **Problema de escalabilidade:** é necessário replicar todo o sistema, mesmo que apenas uma parte exija mais recursos;
- **Implantação lenta e dispendiosa:** cada atualização exige a redistribuição completa da aplicação (Rozanski; Woods, 2025);
- **Baixa resiliência:** erros em um módulo podem comprometer todo o sistema;
- **Obstáculos à inovação tecnológica:** a incorporação de novas ferramentas e linguagens é mais restrita.

Esses elementos fazem com que a arquitetura monolítica seja menos apropriada para projetos que demandam alta disponibilidade, crescimento constante e evolução acelerada.

4.3.2 Desafios da arquitetura de microsserviços

Embora tenha suas vantagens, a implementação da arquitetura de microsserviços acarreta uma maior complexidade na gestão. A grande quantidade de serviços independentes requer práticas avançadas de monitoramento, orquestração e segurança (Newman, 2015).

Entre as limitações mais frequentes, estão:

- **Sobrecarga de comunicação:** a transmissão de informações entre serviços por meio da rede pode causar atrasos e elevar o uso de recursos (Görgülü, 2022);

- **Complexidade da implementação:** para evitar inconsistências, os pipelines de integração e entrega contínua devem ser bem-organizados (Fowler, 2015);
- **Gerenciamento de dados distribuídos:** garantir a consistência entre bancos de dados independentes é um desafio constante;
- **Exigência de maior maturidade técnica:** as equipes devem dominar as práticas de DevOps, automação e monitoramento em tempo real.

Dessa forma, apesar de serem poderosos, os microsserviços exigem investimentos consideráveis em infraestrutura, cultura organizacional e formação técnica.

Tabela 2: Principais desafios das arquiteturas monolítica e de microsserviços

Arquitetura	**Principais Desafios**
Monolítica	Baixa escalabilidade; implantação lenta; alta dependência entre módulos
Microsserviços	Complexidade de orquestração; comunicação distribuída; exige maturidade DevOps

Fonte: Elaborada pelo autor

4.4 Critérios para escolha da arquitetura

Estabelecer a arquitetura de software é uma escolha estratégica que afeta diretamente a escalabilidade, a manutenção e a durabilidade de um sistema. A decisão entre a estratégia monolítica e a de microsserviços deve ser guiada por critérios técnicos e organizacionais claramente estabelecidos (Rozanski; Woods, 2025).

Entre os principais critérios de decisão, destacam-se:

- **Tamanho e complexidade do sistema:** aplicações de pequeno ou médio porte, com requisitos estáveis, geralmente se beneficiam da simplicidade da arquitetura monolítica.

Por outro lado, sistemas grandes, dinâmicos e em constante evolução adotam o modelo de microsserviços (Silva, 2020).

- **Escalabilidade:** projetos que demandam escalabilidade granular, onde apenas módulos específicos precisam se expandir, são mais apropriados para microsserviços (Newman, 2015)
- **Ciclo de desenvolvimento:** os microsserviços proporcionam mais flexibilidade quando é preciso fazer atualizações rápidas e constantes, com equipes atuando simultaneamente. Em contrapartida, em equipes menores, a centralização do monólito pode ser mais eficaz (Fowler, 2015).
- **Infraestrutura e recursos organizacionais:** a implementação de microsserviços requer um nível mais elevado de maturidade nas práticas de DevOps, automação, monitoramento e orquestração. Por outro lado, a arquitetura monolítica exige um menor investimento inicial em infraestrutura e ferramentas de controle (Görgülü, 2022).
- **Tolerância a falhas:** ambientes críticos que exigem alta disponibilidade se beneficiam do isolamento de falhas que os microsserviços oferecem. Por outro lado, aplicações monolíticas têm maior propensão a interrupções em cascata.

Dessa forma, não há um modelo universalmente superior. A escolha deve estar alinhada às necessidades específicas do projeto, às características da equipe e ao contexto organizacional.

5 CONCLUSÃO

A análise comparativa entre arquiteturas monolíticas e de microsserviços mostra que cada uma tem suas vantagens e desvantagens, tornando-as apropriadas para diferentes situações. A arquitetura monolítica é reconhecida pela simplicidade de implementação, custo inicial reduzido e facilidade de gerenciamento em sistemas de pequeno porte. Por outro lado, a arquitetura de microsserviços é mais adequada para aplicações que exigem escalabilidade, resiliência e evolução constante.

A pesquisa destaca que a escolha arquitetural não deve se basear somente em tendências tecnológicas, mas em uma análise cuidadosa das necessidades do sistema e da experiência da equipe envolvida. Projetos menores podem aproveitar a previsibilidade e eficiência da arquitetura monolítica, enquanto sistemas mais complexos encontram nos microsserviços a flexibilidade requerida para enfrentar as constantes mudanças do mercado. (Fowler, 2015; Newman, 2015).

Portanto, é evidente que a seleção da arquitetura deve ser baseada em critérios técnicos, estratégicos e organizacionais para assegurar que a solução seja tanto funcional quanto sustentável durante todo o seu ciclo de vida. Mais do que escolher entre monólito ou microsserviços, é fundamental que as equipes entendam os *trade-offs* de cada estratégia e implementem o modelo que melhor atenda aos objetivos do projeto (Silva, 2020; Görgülü, 2022).

REFERÊNCIAS

FOWLER, Martin. Microservices: a definition of this new architectural term. Disponível em: https://martinfowler.com/articles/microservices.html. Acesso em: 6 jun. 2025.

GÖRGÜLÜ, Emrah. Microservices vs Monolithic Architectures: The Differential Structure Between Two Architectures. MINAR - Journal of Information and Natural Sciences, v. 3, n. 2, p. 45–50, 2022. Disponível em: https://www.minarjournal.com/dergi/microservices-vs-monolithic-architectures-the differential-structure-between-two-architectures20221202031410.pdf. Acesso em: 19 maio 2025.

NEWMAN, Sam. Building microservices: designing fine-grained systems. Beijing: O'Reilly Media, 2015. em: https://book.northwind.ir/bookfiles/building microservices/Building.Microservices.pdf. Acesso em: 6 jun. 2025.

ROZANSKI, Nick; WOODS, Eóin. Software Systems Architecture: Working with Stakeholders Using Viewpoints Perspectives. Disponível em: https://mrce.in/ebooks/Software Fundamentals%20of%20Software%20Architecture.pdf. Acesso em: 19 maio 2025.

SILVA, Hugo Rafael Lopes da. Monolito vs microsserviços: estudo comparativo entre arquiteturas de software. 2020. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) – Instituto de Computação, Universidade Estadual de Campinas, Campinas, 2020. Disponível em: https://www.ic.unicamp.br/~reltech/PFG/2020/PFG-20-20.pdf. Acesso em: 6 jun. 2025.

LEWIS, J.; FOWLER, M. Microservices. 2014. Disponível em: https://martinfowler.com/articles/microservices.html. Acesso em: 1 dez. 2025.

GOMES, Francisco; GABRIEL, Vinicius; ROCHA, Lincoln; REGO, Paulo; TRINTA, Fernando. Observabilidade de desempenho de arquiteturas monolíticas e microsserviços com OpenTelemetry. In: SEMANA DE INFORMÁTICA DO SEMISH (SEMISH), 2024. Anais [...]. Porto Alegre: Sociedade Brasileira de Computação, 2024. Disponível em: https://sol.sbc.org.br/index.php/semish/article/view/29361. Acesso em: 1 dez. 2025.

INTERFACE TECNOLÓGICA