

# Resenha do Artigo: “Microservices” de Martin Fowler

Flávio de Souza Ferreira Junior

22 de agosto de 2025

Esse mês comecei em uma empresa nova que usa arquitetura de microsserviços e, para ser sincero, estou me sentindo um pouco perdido. É tanto serviço pequeno se comunicando que às vezes é difícil entender o fluxo completo das coisas. Para tentar me achar, fui ler o famoso artigo “Microservices” do Martin Fowler (e James Lewis), que meu professor, João Paulo Aramuni, tinha me recomendado. A dica acabou vindo em ótima hora e a leitura foi tipo um mapa para o território novo em que estou pisando.

Até agora, a minha experiência tinha sido basicamente com sistemas “monolíticos”, que é o que o artigo usa como contraponto. E é engraçado como ele descreve exatamente a dor que eu já senti em outros lugares: um sistema gigante, onde tudo está amarrado. Você mexe numa coisinha num canto e, do nada, quebra outra coisa que não tinha nada a ver. Fazer o deploy de uma alteração simples vira um evento, com todo mundo prendendo a respiração.

Aí o artigo vem e explica a ideia dos microsserviços, que é basicamente quebrar esse “monstrão” em vários serviços pequenos e independentes. E algumas coisas fizeram muito sentido pra mim, me ajudando a entender a lógica da empresa nova:

- **Organização por capacidade de negócio:** A ideia de que, em vez de ter um time de “front-end”, um de “back-end” e um de “banco de dados”, você tem um time que é dono do “serviço de pagamentos” de ponta a ponta. Parece que assim as coisas andam mais rápido e o time se sente mais responsável pelo que entrega.
- **Implantação independente:** Essa parte é sensacional. A possibilidade de alterar um serviço pequeno e colocá-lo no ar sem precisar mexer no resto do sistema é um sonho. Adeus, medo do deploy!
- **Liberdade de tecnologia:** Cada serviço pode ser escrito na linguagem ou usar o banco de dados que fizer mais sentido para ele. Isso é muito legal, porque abre a porta para usar a ferramenta certa para o problema certo, em vez de ficar preso a uma única tecnologia para tudo.

Mas o artigo também me deu um banho de água fria e mostrou que não é um mar de rosas. A parte das desvantagens me ajudou a entender por que me sinto perdido às vezes:

- **Complexidade operacional:** Se já é difícil cuidar de um sistema, imagina ter que monitorar, fazer deploy e garantir que dezenas (ou centenas!) de serviços pequenos estão funcionando e se comunicando direito. A necessidade de automação (CI/CD, infraestrutura como código) parece ser gigante.
- **Comunicação entre serviços:** Tudo vira uma chamada de rede, que pode falhar, ser lenta... Tem que programar pensando que o serviço que você precisa chamar pode estar fora do ar a qualquer momento. Isso parece bem mais complicado do que uma simples chamada de função dentro do mesmo projeto.
- **Consistência de dados:** Se o cadastro de um cliente precisa atualizar informações em três serviços diferentes, como você garante que ou tudo funciona ou nada funciona? O artigo fala de “consistência eventual”, um conceito que eu nem conhecia e que já vi que preciso estudar muito.

No final, a minha conclusão é que microsserviços é uma abordagem super poderosa, mas que vem com uma carga enorme de complexidade. Não é uma “bala de prata” que resolve tudo. Entendi que, para um projeto pequeno ou um time que está começando, talvez um monolítico bem organizado ainda seja o melhor caminho. Mas para empresas grandes, com sistemas complexos, as vantagens de agilidade e escalabilidade parecem compensar o desafio.

Para mim, como júnior, foi uma leitura essencial para entender o “porquê” por trás da arquitetura que estou vivenciando e, principalmente, para entender que a escolha certa sempre depende do contexto. Agora me sinto um pouco menos perdido e com um caminho mais claro do que preciso estudar.