

A evolução da engenharia de software trouxe novos paradigmas para lidar com demandas de escalabilidade, flexibilidade e manutenção. A arquitetura de microservices surge como alternativa ao modelo monolítico tradicional.

Aplicações monolíticas têm vantagem na simplicidade inicial, com toda a lógica em um único processo, facilitando testes e implantação. Porém, com o crescimento, surgem dificuldades: necessidade de recompilar todo o sistema para pequenas alterações, escalabilidade apenas global e dificuldade em manter a modularidade.

Já os microservices dividem o sistema em serviços menores e independentes, cada um responsável por uma funcionalidade. Isso permite que cada módulo seja substituído, atualizado e escalado isoladamente. A organização é feita por capacidades de negócio, não por camadas técnicas.

Aqui aplica-se a Lei de Conway: a estrutura do sistema reflete a estrutura da organização. Equipes especializadas por função criam sistemas fragmentados. Equipes multifuncionais, organizadas por domínio de negócio, geram arquiteturas mais coesas.

Outro aspecto é a mentalidade de produto: times são responsáveis por todo o ciclo de vida do software ("you build it, you run it"), aproximando desenvolvedores dos usuários.

Os microservices usam comunicação simples, com protocolos leves como HTTP, preferindo "smart endpoints and dumb pipes". Também possuem governança descentralizada: cada equipe escolhe as tecnologias mais adequadas para seu serviço.

Embora comparada ao SOA, a arquitetura de microservices se diferencia pela simplicidade e descentralização, evitando padrões complexos e governança centralizada.

Microservices representam uma mudança cultural, conectando desenvolvimento ágil, organização de equipes e estratégia empresarial. Oferecem escalabilidade granular, resiliência e alinhamento com os negócios, sendo especialmente atraentes na computação em nuvem.