

Microserviços

- Serviços que atuam de forma independente e através de APIs, tornando eficiente e ágil

Microserviços

- Agilidade e escalabilidade
- Flexibilidade tecnológica
- Muito adotado na atualidade
- Robusto e resistente a falhas

Monolítica

- Bem integrada e difícil de manter
- Sistema fica complexo

Elementos básicos

- Front-end, back-end e banco de dados
- 1º: Camada de Hardware
- 2º: Camada de comunicação: se propaga por todo o programa. É feita por RPCs (troca de mensagens) - não pega os detalhes da comunicação. Métodos: HTTP+REST/THrift (Síncrono) e Message (assíncrono), este último se dividindo em 'pub-sub' e 'solicitação-resposta'
- 3º: Camada de plataforma de aplicação: para auxiliar processos
- 4º: Camada de microserviços: tem todos os microserviços

Vantagens

- Isolamento de falhas: uma falha não afeta outras áreas
- Agilidade: por causa da independência
- Diversidade tecnológica: é adaptável para cada serviço
- Manutenção aprimorada: facilita a identificação e correção de bugs
- Escalabilidade: garantida pela independência

Desafios

- Complexidade: cada parte do sistema deve ser analisada individualmente
- Custo: por exigir uma infraestrutura sofisticada
- Habilidades técnicas: exige uma equipe qualificada
- Tamanho do aplicativo
- Código legado: sistemas antigos podem não se adequar ao modo distribuído

Quando usar?

- Aplicações grandes e complexas
- Para garantir escalabilidade
- Desenvolvimento independente e autonomia de equipes
- Implantação contínua e rápida
- Diversidade tecnológica
- Tolerância a falhas
- Manutenção e evolução contínuas

Tecnologias e ferramentas

- Container: docker
- Orquestração: Kubernetes
- Gateways: Kong, NGINX
- Observabilidade: prometheus, grafana, elk stack