

# Microserviços

## Conceito

Abordagem de desenvolvimento de software que divide uma aplicação em um conjunto de serviços independentes, onde cada serviço possui responsabilidades específicas e pode ser desenvolvido, implantado e escalado de forma independente.

Componente: unidade de software que pode ser substituída e atualizada de forma independente.

## Práticas recomendadas

Definição de interfaces claras entre os serviços.

Separação dos dados por serviço.

Desenvolvimento orientado por domínio.

Uso de ferramentas como contêineres e orquestração de contêineres.

## Considerações

A arquitetura de microservices não é uma solução única para todos os problemas.

Cada organização deve avaliar cuidadosamente se essa abordagem é adequada para suas necessidades.

Importância de considerar benefícios e desafios antes de adotar a arquitetura de microservices.

## Desafios

Gerenciamento da comunicação entre os serviços.

Consistência de dados distribuídos.

Implantação e monitoramento de vários serviços em produção.

## Vantagens

Desenvolvimento e implantação independentes dos serviços, permitindo maior agilidade.

Uso de diferentes tecnologias e linguagens de programação em cada serviço

Escalabilidade granular de serviços específicos.

Facilita a colaboração e a evolução rápida em equipes distribuídas.

## Características

### Desacoplamento

Os microservices são projetados para serem independentes uns dos outros, permitindo que sejam desenvolvidos, implantados e alterados sem afetar o restante do sistema.

Cada serviço tem seu próprio contexto delimitado e pode ser substituído ou atualizado sem impactar outros serviços.

### Elasticidade e Resiliência

Os microservices facilitam a escalabilidade horizontal, permitindo aumentar ou diminuir o número de instâncias de um serviço de acordo com a demanda.

A falha em um serviço não afeta necessariamente todo o sistema, pois outros serviços podem continuar funcionando normalmente.

### Comunicação

Geralmente, a comunicação entre os microservices ocorre por meio de APIs, utilizando protocolos como HTTP/REST ou mensagens assíncronas.

É comum o uso de mecanismos como filas de mensagens, barramentos de eventos ou descoberta de serviços para facilitar a comunicação e o gerenciamento das interações entre os serviços.

### Implantação e Gerenciamento

O uso de contêineres, como Docker, facilita a implantação e o isolamento dos microservices.

A orquestração de contêineres, por meio de ferramentas como Kubernetes, simplifica o gerenciamento, a escalabilidade e a resiliência dos serviços.

### Monitoramento

É essencial monitorar e coletar dados sobre o desempenho e o comportamento dos microservices.

Ferramentas de monitoramento e registro de logs são utilizadas para detectar problemas, identificar gargalos e tomar ações corretivas.

### Cultura DevOps

A arquitetura de microservices está fortemente alinhada com a cultura DevOps, promovendo colaboração entre desenvolvimento e operações.

As equipes são responsáveis pela construção, implantação e operação dos serviços que desenvolvem, fomentando um ciclo de feedback contínuo.

### Testabilidade e Manutenibilidade

Os microservices facilitam a realização de testes automatizados, pois cada serviço pode ser testado individualmente.

Atualizações e manutenções se tornam mais gerenciáveis, uma vez que alterações em um serviço específico não exigem uma implantação completa do sistema.

Autor: Heitor Rodrigues