

Exercício - Introdução a Microserviços

Nome: Guilherme Franco Batista

GRR20211603

Descreva:

1. Problemas de uma arquitetura monolítica de aplicações
2. O que são microserviços
3. Diferença entre arquitetura monolítica e microserviços
4. Prós e Contras Monolito
5. Prós e Contras Microserviços
6. Quando usar Monolito e quando usar Microserviços
7. Ecossistema de microserviços, bem como suas camadas

1) A arquitetura monolítica possui alguns problemas:

- O código é muito grande
- Produtividade no desenvolvimento é baixa: IDEs não rodam bem
- Container Web sobrecarregado pelo tamanho do monolito
- Implantação contínua difícil: reimplantar a aplicação inteira toda vez
- Dificuldade para escalar: somente em uma dimensão, o aumento no volume de dados é complicado.
- Requer um compromisso de longo prazo com a tecnologia

2) No monolito temos uma aplicação única, onde há dependência entre serviços de uma aplicação, os módulos compartilham recursos, processamento, memória, banco de dados e arquivos.

3) Os microserviços são mais escaláveis e dinâmicos, você pode escolher as partes/módulos que você deseja escalar, enquanto no monolito você precisaria escalar toda a aplicação.

4) Vantagens

- Estruturação única, tudo segue um mesmo padrão
- Poucos recursos tecnológicos, como por exemplo apenas o servidor de aplicação e o de banco de dados

- Não necessita de muitas pessoas cuidando
- Não precisa de microserviços
- Acesso a módulos distintos sem precisar de integração

Contras

- Aumento da complexidade ao longo do tempo
- Alto acoplamento: Altíssima dependência de componentes de código
- Escabilidade limitada
- Pouca flexibilidade
- Baixa manutenibilidade (mais difícil encontrar o erro)
- Confiabilidade: se houver erro em algum módulo, pode afetar a disponibilidade afeta todo o aplicativo

5) • Benefícios

- Permite entrega e implantação contínuas de aplicações grandes e complexas
- Manutenibilidade melhorada: cada serviço é pequeno e fácil de entender
- Testabilidade melhorada: serviços podem ser implantados independentemente
- Melhora o desenvolvimento: cada time é responsável por um ou vários serviços
- Elimina o comprometimento de longo prazo com a tecnologia
- Problemas
 - Desenvolvedores devem se adaptar ao desenvolvimento distribuído
 - Requisições que abrangem vários serviços podem ser difíceis de implementar
 - Teste de integração entre serviços é mais complicado

- Complexidade da implantação aumenta, deve-se tratar os aspectos de vários serviços sendo geridos

- Aumento no consumo de memória: troca n instâncias de monolitos por $n \times m$ instâncias de serviços. Se cada serviço roda na sua própria VM, o overhead é ainda maior

6) O Monolito é recomendado para aplicação que não sejam complexas e que não precisem estar divididas entre vários microsserviços, já os Microsserviços são recomendados para projetos que vão ser complexos e vão ter uma escalabilidade alta.

7) De acordo com o modelo criado pela engenheira de software Susan J. Fowler, o ecossistema de microsserviços está dividido em quatro camadas:

1 – Hardware:

Os hardwares são os computadores e equipamentos nos quais os microsserviços são armazenados e executados.

2 – Comunicação:

A comunicação engloba todo o conjunto responsável por permitir a interação entre os microsserviços na aplicação. Ela é composta por elementos chave como os endpoints de API, Service Discovery, Service Registry e Load Balancers.

3 – Plataforma de aplicações;

A plataforma de aplicações, engloba todas as ferramentas que são independentes de um microsserviço. Essas ferramentas devem ser construídas e dispostas de forma que o time de desenvolvimento não tenha que se preocupar com nada além da lógica da própria aplicação.

4 – Microsserviços;

A ideia é que os microsserviços sejam totalmente abstraídos dos demais componentes que foram mencionados acima. A camada de microsserviços é composta apenas pelo código e pelas configurações específicas que devem ser aplicadas sobre ele, para que seja possível entregar as funcionalidades que foram idealizados para os microsserviços.