

Infnet

Gustavo Pimentel

Desenvolvimento de Software Ágil e Escalável com Microserviços

AT

1. O que são microserviços? Quando microserviços são uma boa opção? Quando não são?

Microserviços são serviços pequenos,
independentes e modulares que são

implementados no software em vez de um monólito, que seria um grande serviço que lida com tudo de uma aplicação.

Microserviços são uma opção boa pra aplicações grandes, com muitas funcionalidades e que serão trabalhadas por múltiplas pessoas ou equipes. Nesses casos eles são muito úteis de implementar pois facilitam a questão da escalabilidade e independência do desenvolvimento de cada parte da aplicação.

No caso de aplicações bem simples e pequenas, com poucos recursos, ou que são certas de que não vão ser escaladas, pode ser uma opção melhor fazer de forma monolítica, já que pode ser mais simples do ponto de vista estrutural.

2. Por que o método DDD de projeto de software ajuda na elaboração de microserviços?

DDD (Domain Driven Design) facilita muito na elaboração de softwares com arquitetura de microserviços porque ele foca na separação de desenvolvimento e independência de cada domínio, da mesma forma que os microserviços focam na modularização dos serviços. Dessa forma cada microserviço pode estar focado em um

domínio e uma entrega diferente, com bastante clareza na responsabilidade de cada microsserviço.

3. O que é um Message Broker?

Um message broker é um programa que age como intermediário entre a aplicação e os microsserviços, permitindo que eles se comuniquem entre si através da troca de mensagens (por isso message broker), traduzindo a mensagem entre as linguagens e simplificando-a, removendo os detalhes desnecessários que poderiam atrapalhar a comunicação.

4. Explique com detalhes o que é o Padrão Retry Pattern e Circuit Breaker.

Ambos são técnicas usadas para aumentar a tolerância das aplicações a possíveis erros.

Retry Pattern: Nessa técnica, a aplicação fica tentando refazer alguma operação que deu erro algumas vezes em pequenos intervalos de tempo. Esse padrão funciona bem para algumas operações que podem falhar em casos de congestionamento de rede ou um servidor congestionado, porque são problemas pontuais, que fazem com que haja uma grande chance do erro não acontecer

eventualmente, se continuar tentando.

Circuit Breaker: O Circuit Breaker funciona em casos que o Retry Pattern não seria apropriado, indicando que há um problema mais grave no serviço sendo chamado, que não seria resolvido apenas tentando várias vezes. Nesses casos o Circuit Breaker detecta a grande quantidade de falhas na operação, e impede a aplicação de continuar tentando, evitando assim que o serviço que já está falhando seja ainda mais sobrecarregado e dando tempo para ele se recuperar, ou avisando os desenvolvedores para eles verificarem o que está errado.