

EXERCÍCIOS DO MÓDULO 06: ARQUITETURA DE SOFTWARE O MONÓLITO

Samuel Ferreira Machado¹

EXERCÍCIO 1 - MONÓLITOS VS MICROSERVIÇOS - ESCALABILIDADE

Explique, em termos de Escalabilidade, as vantagens e desvantagens das arquiteturas de software de Monólitos e Microserviços.

Arquitetura de Monólitos:

Vantagens em termos de escalabilidade:

- a) Implantação Simples: Devido à sua natureza de ser um único aplicativo, a implantação de um monólito é relativamente simples em comparação com a arquitetura de microserviços. Isso pode resultar em uma implantação mais rápida e eficiente.
- b) Menor Overhead de Comunicação: Dentro de um monólito, os componentes se comunicam diretamente uns com os outros, sem a necessidade de comunicação através de redes ou APIs. Isso reduz o overhead de comunicação e pode resultar em tempos de resposta mais rápidos em comparação com uma arquitetura de microserviços.

Desvantagens em termos de escalabilidade:

- a) Escalabilidade Limitada: Os monólitos têm uma escalabilidade limitada, pois todo o aplicativo é implantado como uma única unidade. Isso significa que, se uma parte específica do aplicativo requer mais recursos, é necessário escalar o aplicativo inteiro, mesmo que outras partes não precisem dessa escalabilidade adicional.
- b) Dificuldade de Manutenção: À medida que um monólito cresce em tamanho e complexidade, a manutenção pode se tornar desafiadora. Alterar ou corrigir um componente pode afetar outros componentes, resultando em um ciclo de desenvolvimento mais lento e maior risco de introdução de erros.

Arquitetura de Microserviços:

Vantagens em termos de escalabilidade:

¹ Aluno do Curso Especialista Back-End Java da EBAC.

- a. Escalabilidade Granular: A arquitetura de microsserviços permite escalar componentes individuais de forma independente. Isso significa que é possível dimensionar apenas os serviços que exigem mais recursos, sem a necessidade de dimensionar todo o sistema. Isso proporciona uma escalabilidade mais eficiente e econômica.
- b. Tolerância a Falhas: Os microsserviços são projetados para serem isolados uns dos outros. Se um serviço falhar, os outros serviços podem continuar funcionando normalmente. Isso aumenta a tolerância a falhas e a disponibilidade geral do sistema.

Desvantagens em termos de escalabilidade:

- a. Complexidade de Orquestração: Com múltiplos microsserviços interagindo entre si, a complexidade de orquestrar e coordenar a comunicação entre esses serviços aumenta. Isso pode exigir o uso de ferramentas de orquestração e pode adicionar complexidade à infraestrutura do sistema.
- b. Latência de Rede: Os microsserviços normalmente se comunicam por meio de chamadas de rede, o que pode introduzir latência adicional em comparação com uma arquitetura monolítica, onde a comunicação ocorre internamente. Em um sistema distribuído com microsserviços, é necessário levar em consideração a latência de rede ao projetar a escalabilidade do sistema.

EXERCÍCIO 2 - MONÓLITOS VS MICROSERVIÇOS - DISPONIBILIDADE

Explique, em termos de Disponibilidade, as vantagens e desvantagens das arquiteturas de software de Monólitos e Microsserviços.

Arquitetura de Monólitos:

Vantagens em termos de disponibilidade:

- a) Menor complexidade de implantação: Com um único aplicativo, a implantação de um monólito pode ser mais simples em comparação com microsserviços, o que pode reduzir o tempo de inatividade durante o processo de implantação.
- b) Maior coesão: Todos os componentes de um monólito estão no mesmo ambiente, o que pode facilitar a detecção e resolução de problemas de disponibilidade. É mais fácil garantir que o sistema esteja funcionando corretamente como um todo.

Desvantagens em termos de disponibilidade:

- a) Alto risco de falha completa: Se um componente do monólito falhar, todo o sistema pode ficar inoperante. Isso pode resultar em uma menor disponibilidade geral, já que um único ponto de falha pode afetar todo o aplicativo.
- b) Dificuldade de escalabilidade: A falta de modularidade em um monólito pode dificultar a escalabilidade, especialmente quando há picos de tráfego em partes específicas do sistema. A escalabilidade granular é limitada e pode afetar a disponibilidade em momentos de carga intensa.

Arquitetura de Microserviços:

Vantagens em termos de disponibilidade:

- a) Tolerância a falhas: Com microserviços isolados, se um serviço falhar, outros serviços podem continuar funcionando normalmente, aumentando a disponibilidade geral do sistema.
- b) Escalabilidade granular: Com microserviços, é possível dimensionar apenas os serviços específicos que exigem mais capacidade, enquanto os outros serviços permanecem inalterados. Isso permite uma melhor adaptação à demanda e maior disponibilidade em momentos de pico de tráfego.

Desvantagens em termos de disponibilidade:

- a) Complexidade de gerenciamento: A arquitetura de microserviços introduz uma complexidade adicional no gerenciamento do ambiente de produção. A coordenação e a orquestração de vários serviços podem ser desafiadoras, aumentando o risco de erros e interrupções.
- b) Dependência de rede: Os microserviços se comunicam por meio de chamadas de rede, o que aumenta a dependência da rede. Problemas de rede, como latência ou falhas, podem afetar a disponibilidade dos serviços e, consequentemente, do sistema como um todo.

EXERCÍCIO 3 - MONÓLITOS VS MICROSERVIÇOS - CONSISTÊNCIA

Explique, em termos de Consistência, as vantagens e desvantagens das arquiteturas de software de Monólitos e Microserviços.

Arquitetura de Monólitos:

Vantagens em termos de consistência:

- a) Coerência de dados: Dentro de um monólito, os componentes compartilham o mesmo banco de dados, o que facilita a manutenção da consistência dos dados. Alterações em diferentes partes do sistema são aplicadas de forma mais simples e direta.
- b) Transações ACID: Com um banco de dados centralizado, é possível usar transações ACID (Atomicidade, Consistência, Isolamento e Durabilidade) para garantir a consistência dos dados. Isso significa que as operações de gravação e leitura são atomicamente consistentes e podem ser revertidas em caso de falha.

Desvantagens em termos de consistência:

- a) Acoplamento rígido: Em uma arquitetura monolítica, as diferentes partes do sistema estão fortemente acopladas, o que pode dificultar a manutenção da consistência quando ocorrem mudanças em um componente específico. Atualizações em um módulo podem afetar outros módulos de maneira imprevisível, resultando em problemas de consistência.
- b) Dificuldade de escalabilidade: Quando se trata de escalabilidade horizontal, um monólito pode ter dificuldades para manter a consistência dos dados à medida que mais instâncias do aplicativo são adicionadas. A sincronização de dados entre as instâncias pode ser complexa e impactar a consistência global do sistema.

Arquitetura de Microsserviços:

Vantagens em termos de consistência:

- a) Consistência por serviço: Cada microsserviço possui seu próprio banco de dados dedicado e responsabilidade sobre os dados que manipula. Isso permite que cada serviço garanta sua própria consistência interna, mantendo a integridade dos dados dentro de seu domínio.
- b) Consistência eventual: Em vez de depender de transações ACID, a arquitetura de microsserviços pode adotar uma abordagem de consistência eventual. Isso significa que, embora haja algum atraso na propagação das alterações entre

serviços, a consistência é alcançada ao longo do tempo, permitindo maior escalabilidade e desempenho.

Desvantagens em termos de consistência:

- a) Complexidade de coordenação: Com a divisão do sistema em vários microsserviços, a coordenação de transações distribuídas e a manutenção da consistência entre serviços podem se tornar complexas. A necessidade de garantir a consistência global dos dados requer a implementação de estratégias de controle de concorrência e coordenação eficazes.
- b) Eventualmente consistente: A consistência eventual pode significar que, por um curto período, os dados entre serviços podem estar inconsistentes. Isso pode afetar operações que dependem de dados atualizados em tempo real, exigindo mecanismos adicionais para lidar com a eventual consistência, como estratégias de compensação e reconciliação.

EXERCÍCIO 4 - MONÓLITOS VS MICROSERVIÇOS - PERFORMANCE

Explique, em termos de Performance, as vantagens e desvantagens das arquiteturas de software de Monólitos e Microsserviços.

Arquitetura de Monólitos:

Vantagens em termos de performance:

- a) Baixa latência de comunicação interna: Em um monólito, os componentes se comunicam internamente, sem a necessidade de comunicação por meio de redes ou APIs. Isso resulta em uma menor latência de comunicação, o que pode melhorar o desempenho em comparação com a arquitetura de microsserviços.
- b) Otimização do código: Com todos os componentes no mesmo aplicativo, é possível otimizar o código para obter um desempenho mais eficiente. A otimização pode ser feita em nível de código e dependências, aproveitando melhorias de desempenho em toda a aplicação.

Desvantagens em termos de performance:

- a) Escalabilidade limitada: Os monólitos têm uma escalabilidade limitada, pois todo o aplicativo é implantado como uma única unidade. Isso pode resultar em

gargalos de desempenho e falta de escalabilidade para lidar com cargas de trabalho crescentes.

- b) Dificuldade de isolamento de recursos: Com todos os componentes compartilhando os mesmos recursos, como memória e CPU, é mais difícil isolar e gerenciar os recursos de forma eficiente. Isso pode levar a problemas de contenção e afetar negativamente o desempenho do sistema.

Arquitetura de Microsserviços:

Vantagens em termos de performance:

- a) Escalabilidade granular: Com microsserviços, é possível dimensionar apenas os serviços específicos que requerem mais recursos, sem a necessidade de escalar todo o sistema. Isso permite uma melhor adaptação à demanda e um melhor desempenho em momentos de carga intensa.
- b) Melhor isolamento de recursos: Cada microsserviço é executado em seu próprio ambiente, permitindo um melhor isolamento de recursos, como memória e CPU. Isso evita problemas de contenção e garante um desempenho mais previsível e estável.

Desvantagens em termos de performance:

- a) Overhead de comunicação: A comunicação entre microsserviços geralmente ocorre por meio de chamadas de rede, o que pode adicionar um overhead de comunicação e resultar em maior latência em comparação com a arquitetura de monólitos. Isso pode afetar o desempenho do sistema, especialmente em casos em que a latência é crítica.
- b) Complexidade de gerenciamento: Com a divisão do sistema em vários microsserviços, o gerenciamento e a orquestração dos serviços podem se tornar complexos. A coordenação entre os serviços pode introduzir atrasos e sobrecarga adicional, afetando o desempenho geral do sistema.

EXERCÍCIO 5 - MONÓLITOS VS MICROSERVIÇOS - TESTABILIDADE

Explique, em termos de Testabilidade, as vantagens e desvantagens das arquiteturas de software de Monólitos e Microsserviços.

Arquitetura de Monólitos:

Vantagens em termos de testabilidade:

- a) Unidade de teste coesa: Em um monólito, todos os componentes estão no mesmo aplicativo, o que facilita a realização de testes de unidade. É mais fácil isolar e testar as funcionalidades do sistema como um todo, garantindo uma cobertura abrangente dos testes.
- b) Testes de integração simplificados: Como todos os componentes estão acoplados internamente, a execução de testes de integração pode ser mais fácil em comparação com a arquitetura de microsserviços. É possível simular cenários de integração mais complexos sem a necessidade de gerenciar comunicação e dependências entre diferentes serviços.

Desvantagens em termos de testabilidade:

- a) Dependência de ambiente completo: Para testar um componente específico de um monólito, muitas vezes é necessário executar o ambiente completo do aplicativo. Isso pode aumentar a complexidade dos testes e torná-los mais lentos de serem executados.
- b) Dificuldade de isolamento de falhas: Como todos os componentes estão intimamente acoplados, é mais difícil isolar e identificar falhas em um único componente durante os testes. Um problema em um componente pode afetar outros, dificultando a identificação e resolução de problemas específicos.

Arquitetura de Microsserviços:

Vantagens em termos de testabilidade:

- a) Testes de unidade independentes: Cada microsserviço pode ser testado individualmente, permitindo a realização de testes de unidade mais isolados e específicos para cada serviço. Isso facilita a identificação e resolução de problemas em um serviço específico.
- b) Testes de integração controlados: Com microsserviços, a comunicação entre serviços ocorre por meio de interfaces definidas e bem delimitadas, o que torna os testes de integração mais controlados. É possível testar a integração entre serviços de forma mais granular e identificar problemas mais facilmente.

Desvantagens em termos de testabilidade:

- a) Complexidade de cenários de teste: A arquitetura de microsserviços pode introduzir uma complexidade adicional na definição e execução de cenários de teste abrangentes que envolvam a interação de vários serviços. A coordenação e a simulação de diferentes casos podem ser mais desafiadoras.
- b) Ambiente de teste distribuído: Os microsserviços são distribuídos e executados em ambientes independentes, o que pode dificultar a configuração e o gerenciamento do ambiente de teste. É necessário considerar a infraestrutura e as ferramentas adequadas para criar um ambiente de teste eficiente.

REFERÊNCIAS

Arquitetura de microsserviços X arquitetura monolítica: 7 diferenças.

Disponível em: <https://blog.vinco.com.br/arquitetura-de-microsservicos-x-arquitetura-monolitica/>. Acesso em 07 jul. 2023.

Microsserviços versus arquitetura monolítica.

Disponível em: <https://www.atlassian.com/br/microservices/microservices-architecture/microservices-vs-monolith>. Acesso em 07 jul. 2023.

Microsserviços x arquitetura monolítica: entenda a diferença.

Disponível em: <https://viceri.com.br/insights/microsservicos-x-arquitetura-monolitica-entenda-a-diferenca/>. Acesso em 07 jul. 2023.

Microsserviços: conceito, vantagens e desvantagens dessa arquitetura.

Disponível em: <https://www.supero.com.br/blog/microsservicos-conceito-vantagens-e-desvantagens-desse-tipo-de-arquitetura/>. Acesso em 07 jul. 2023.