

Disciplina: Arquitetura de Backend

Professor: Marco Mendes

Exercícios Unidade 04

A arquitetura de microsserviços tem ganhado destaque como uma abordagem eficaz para projetar e desenvolver sistemas complexos. Neste cenário, imagine que você é um arquiteto de software renomado, responsável por projetar uma solução baseada em microsserviços para um provedor de streaming de música. Sua tarefa é criar uma arquitetura que possibilite a entrega contínua de música aos usuários e, ao mesmo tempo, garantir escalabilidade, flexibilidade e alta disponibilidade.

Cenário:

A plataforma de streaming musical é composta por várias funcionalidades, incluindo:

1. **Catálogo de Músicas:** Armazena informações sobre músicas, álbuns e artistas disponíveis na plataforma.
2. **Recomendações:** Gera recomendações personalizadas para os usuários com base em seus gostos musicais.
3. **Autenticação e Perfis de Usuário:** Gerencia a autenticação dos usuários e suas preferências musicais.
4. **Reprodução de Música:** Responsável por transmitir músicas aos usuários, controlando a reprodução e a qualidade do áudio.

Desafio:

Você precisa projetar uma arquitetura de microsserviços que permita a comunicação eficiente e a entrega contínua de música aos usuários. Além disso, você deve incorporar padrões de microsserviços para melhorar a escalabilidade, a resiliência e a manutenibilidade do sistema.

Problemas:

1. **Escalabilidade de Reprodução de Música:** Como você projetaria a arquitetura para garantir que a reprodução de música possa ser escalada horizontalmente para lidar com um aumento repentino no número de usuários ativos?
2. **Consistência de Dados no Catálogo de Músicas:** Explique como você lidaria com a consistência de dados no catálogo de músicas, considerando a atualização de informações sobre artistas, álbuns e músicas.
3. **Resiliência em Recomendações:** Descreva como você implementaria o padrão "Circuit Breaker" para garantir que a funcionalidade de recomendações permaneça disponível, mesmo diante de falhas em sistemas subjacentes.
4. **Integração de Perfis de Usuário:** Como você projetaria a integração entre o sistema de autenticação e o perfil de usuário para garantir uma experiência personalizada e segura para os usuários?
5. **Entrega Contínua de Música:** Explique como você implementaria o padrão "Event Sourcing" para garantir que a entrega contínua de música aos usuários seja rastreável e confiável.

Questões:

1. **Padrões de Microserviços:** Identifique e explique como você aplicaria o padrão de "Service Discovery" em sua arquitetura de microserviços para facilitar a comunicação entre os diversos serviços.
2. **Comparação de Abordagens:** Compare a abordagem de "Database Per Service" com a de "Shared Database" em relação à consistência de dados e à manutenibilidade. Explique em que cenários cada abordagem seria mais vantajosa.
3. **Segurança na Comunicação:** Discorra sobre como você garantiria a segurança da comunicação entre os microserviços, especialmente quando dados sensíveis, como informações de autenticação, precisam ser transmitidos.
4. **Resiliência e Tolerância a Falhas:** Além do padrão "Circuit Breaker", mencione outro padrão de resiliência que você incorporaria em sua arquitetura e explique como ele melhoraria a disponibilidade e a estabilidade do sistema.
5. **Impacto nos Usuários:** Analise como a adoção da arquitetura de microserviços e dos padrões discutidos afetaria a experiência dos usuários no serviço de streaming musical. Considere aspectos como desempenho, personalização e disponibilidade.

Este exercício tem como objetivo avaliar sua compreensão dos conceitos de arquitetura de microserviços e padrões associados. Certifique-se de oferecer respostas detalhadas, embasadas e que abordem os aspectos solicitados em cada pergunta.

Problemas e Soluções:

1. Escalabilidade de Reprodução de Música:

Solução: A arquitetura deve ser projetada para suportar a escalabilidade horizontal da reprodução de música. Isso pode ser alcançado por meio do uso de um balanceador de carga que distribui as solicitações dos usuários entre várias instâncias do serviço de reprodução. Além disso, o uso de uma arquitetura de microserviços permite a replicação desses serviços para lidar com um aumento no tráfego. O Kubernetes pode ser utilizado para gerenciar o escalonamento automático.

2. Consistência de Dados no Catálogo de Músicas:

Solução: Para garantir a consistência de dados, é recomendado aplicar o padrão "Database Per Service". Cada microserviço, como o Catálogo de Músicas, teria sua própria base de dados dedicada. A comunicação entre microserviços aconteceria através de APIs, garantindo a separação de preocupações e minimizando a complexidade de manter dados consistentes.

3. Resiliência em Recomendações:

Solução: A implementação do padrão "Circuit Breaker" é crucial para a resiliência em recomendações. Quando o serviço de recomendações detecta falhas em sistemas subjacentes, ele interrompe temporariamente as requisições a esses sistemas. Isso evita o impacto negativo em toda a aplicação devido a sistemas lentos ou indisponíveis.

4. Integração de Perfis de Usuário:

Solução: O padrão "API Gateway" pode ser usado para integrar o sistema de autenticação e o perfil do usuário. O API Gateway atua como um ponto de entrada único para os microserviços e pode lidar com a autenticação dos usuários antes de encaminhar as requisições para os microserviços relevantes. Isso garante uma experiência personalizada e segura.

5. Entrega Contínua de Música:

Solução: O padrão "Event Sourcing" pode ser aplicado para garantir a entrega contínua de música. Cada reprodução de música é tratada como um evento que é registrado e pode ser rastreado. Isso permite a recuperação de informações em caso de falha e facilita a auditoria das atividades.

Questões:

1. Padrões de Microsserviços:

O padrão de "Service Discovery" é fundamental para uma arquitetura de microsserviços. Ele permite que os microsserviços encontrem e se comuniquem entre si dinamicamente. Uma ferramenta como o Consul ou o Eureka pode ser usada para implementar esse padrão, fornecendo um registro centralizado de serviços disponíveis.

2. Comparação de Abordagens:

- **Database Per Service:** Separação de bases de dados para cada serviço, garantindo autonomia e evitando problemas de consistência. Mais complexidade de gerenciamento.
- **Shared Database:** Uso de uma única base de dados para todos os serviços. Pode gerar problemas de escalabilidade e consistência, mas pode ser mais simples de gerenciar em pequenas aplicações.
-

3. Segurança na Comunicação:

A segurança da comunicação entre os microsserviços pode ser garantida através do uso de protocolos de criptografia, como HTTPS, e da implementação de autenticação e autorização robustas, como OAuth 2.0. O uso de tokens de acesso e certificados SSL/TLS pode assegurar a confidencialidade e autenticidade dos dados transmitidos.

4. Resiliência e Tolerância a Falhas:

Além do "Circuit Breaker", outro padrão importante é o "Bulkhead". Ele envolve a separação de componentes em compartimentos isolados, de modo que falhas em um compartimento não afetem outros. Isso melhora a resiliência, uma vez que uma falha em um compartimento não sobrecarrega os recursos disponíveis para outros compartimentos.

5. Impacto nos Usuários:

A adoção da arquitetura de microsserviços e dos padrões discutidos pode trazer vários benefícios para os usuários. A escalabilidade permitiria que mais usuários desfrutassem do serviço simultaneamente, enquanto a resiliência garantiria que a plataforma permanecesse disponível mesmo em face de falhas. Além disso, a personalização das recomendações e a entrega contínua de música melhorariam a experiência do usuário, mantendo-os envolvidos e satisfeitos.