Curso: Arquitetura de software distribuído – Oferta 12

Alunos: Geysa Fernandes, Vinicius Alberto, Wellington Silva

**AuditoriaWebAPI**

1. **Organize APIs ao longo de recursos:** 
   1. Cada API foi organizada de acordo com o núcleo de negócio da qual ela trata. Por exemplo a AuditoriaWebAPI trata apenas do processo de auditoria e possui apenas duas rotas de inclusão e de recuperação de registros de auditoria, que são distintos entre si pelo verbo HTTP utilizado.
2. **Padronize suas APIs:**
   1. Na AuditoriaWebAPI as URIs foram padronizadas para fazerem bom uso dos verbos HTTP a fim de denotarem o resultado obtido de cada chamada.
3. **Evite APIs anêmicas:**
   1. A AuditoriaWebAPI foi organizada de forma a possuir uma cada de regras de negócio. Entretanto, a API só possui as funções de inserir e recuperar não existindo regras de negócio implementadas.
4. **Crie APIs simples:**
   1. As rotas criadas na AuditoriaWebApi são bem simples, sendo utilizados os verbos HTTP para diferenciar o funcionamento das rotas. Além da documentação Swagger.
5. **Considere a atualização em lote para operações complexas:** 
   1. Não são feitas operações complexas na API, apenas inserção e recuperação de dados.
6. **Se você precisar receber data e horas na sua API, use o padrão ISSO 8601:** 
   1. A AuditoriaWebAPI utiliza a Data e Hora fornecidas pela chamada.
7. **Documente sua API:**
   1. Foi utilizado o padrão OpenAPI na documentação utilizando-se o Swagger.
8. **Use protocolo HTTPS/SSL. Sempre:**
   1. A API utiliza o protocolo de segurança HTTP/SSL.
9. **Versione sua API:**
   1. As rotas expostas na API foram versionadas na versão V1, não existindo rotas em outra versão.
10. **Estabeleça paginação para coleções com grande volume de dados:**
    1. Foi feita uma paginação para o retorno de volume de dados muito grandes.
11. **Use corretamente os códigos de retorno HTTP:**
    1. Os verbos HTTP foram corretamente utilizados para diferenciar a função oferecida por cada rota.

**Qual a sua percepção sobre micro serviços:**

A arquitetura de micro serviços vem sendo adotada cada vez mais em empresas a fim de organizar os recursos e promover a independência entre as aplicações. Um micro serviço é caracterizado por ser uma aplicação pequena e independente que expõe funcionalidades sobre a manipulação de algum recurso. Um micro serviço tem um ambiente de implantação próprio e uma base de dados própria, dessa forma pode-se implementar e expor um micro serviço sem a necessidade de outros micro serviços ou aplicações.

Entre as principais vantagens da utilização de micro serviços estão:

1. Cada serviço pode ser implantado independentemente de outros serviços. Na arquitetura monolítica a aplicação poderia ser implantada apenas quando todas as partes dependentes estivessem desenvolvidas, com o micro serviço cada serviço pode ser implantado independentemente e estar disponível para chamadas sem dependência de outras aplicações.
2. Cada serviço pode ser escalonado independentemente de outros serviços através da duplicação ou particionamento. Visto que o micro serviço é uma aplicação independente ela pode ser publicada facilmente em várias instancias de acordo com a necessidade.
3. A arquitetura de micro serviços elimina compromissos de longo prazo com a pilha tecnológica, visto que cada micro serviço pode ser desenvolvido em sua tecnologia e arquitetura independente. Podendo ainda haver comunicação entre aplicações desenvolvidas em tecnologia antiga e tecnologia nova.

Algumas das desvantagens da utilização de micro serviços são:

1. As equipes (squads) que atuam em um micro serviço devem lidar com uma complexidade adicional com o desenvolvimento de uma aplicação completa para cada micro serviço, visto que cada micro serviço deve ter sua própria arquitetura, teste unitários, banco de dados etc.
2. A arquitetura de micro serviços introduz uma complexidade operacional significativa, visto que mais elementos, às vezes com múltiplas instancias, devem ser gerenciados no servidor.
3. A implantação de funcionalidades de abrangem vários micro serviços deve ser acordada, coordenada e sincronizada, às vezes requerendo a ação de várias equipes.