



# Microservices

Arquitetura de Micro Serviços

Matheus Santana da Silva

07/09/2020

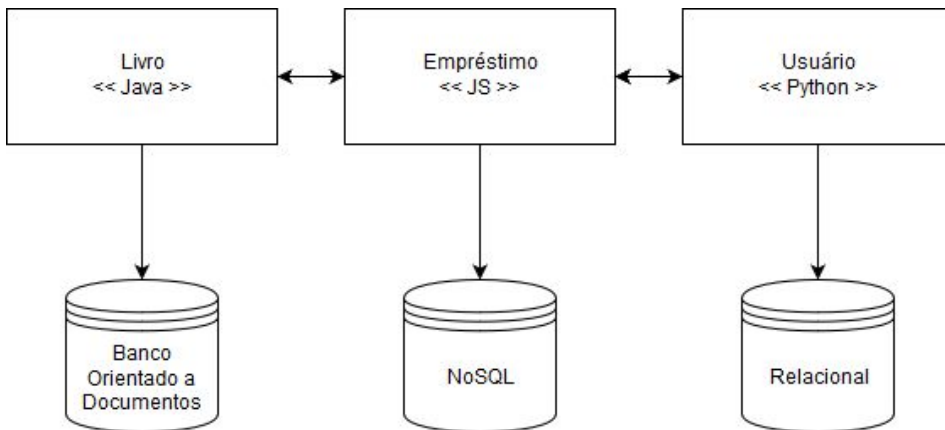


# O que são Microservices?

- Microservices podem ser definidos como pequenos serviços autônomos que trabalham juntos;
- Uma outra definição possível seria uma coleção de serviços que tem como características principais:
  - Facilidade de manutenção e testes, uma vez que os serviços são independentes entre si;
  - Deploy Independente;
  - Alta coesão e baixo acoplamento;
    - São altamente especializados em realizar tarefas específicas;
    - Não dependem uns dos outros para operar.
  - São organizados em volta das capacidades do negócio.

# Quais os Benefícios?

- Completa liberdade em relação a tecnologia;
  - Como um mesmo sistema é composto por vários serviços independentes, é possível implementar cada um dos serviços que compõe o sistema com uma tecnologia diferente, proporcionando a equipe de desenvolvimento liberdade de escolher a melhor tecnologia para determinada tarefa;



# Quais os Benefícios?

- Escalabilidade
  - Dentro de um sistema monolítico só é possível escalar o sistema como um todo, mesmo que o problema da aplicação se concentre em apenas uma tarefa específica. Com a arquitetura de microservices não temos mais este paradigma, uma vez que trabalhando com serviços menores é possível escalar apenas as partes do sistema que realmente precisam ser escalonadas .





# Quais os Benefícios?

- **Resiliência**
  - Caso uma instância do sistema caia ou apresente falhas, o restante do sistema não é afetado;
- **Facilidade no Deploy**
  - Em uma aplicação monolítica tradicional, existe um alto risco inerente a atividade de deploy, uma vez que para que uma mudança seja efetivada é necessário o deploy da aplicação como um todo, aumentando o consideravelmente o risco de falha.
  - Com uma arquitetura de micro serviços, é possível alterar um serviço em específico realizando o deploy do mesmo de forma independente, sem afetar os outros serviços em produção, diminuindo bastante o risco inerente desta atividade.
- **Alinhamento Organizacional**
  - Com um sistema construído em uma arquitetura de micro serviços é possível separar melhor os times de desenvolvimento de forma que estes trabalhem em serviços específicos, que utilizem tecnologias específicas que possam extrair o máximo da capacidade de cada integrante do time, além de limitar o número de pessoas que trabalham em um mesmo código.



# Quais os Benefícios?

- Reusabilidade
  - Uma arquitetura de microservices, assim como um sistema distribuído ou uma arquitetura voltada a serviços, tem seu foco na reusabilidade do código gerado, ou seja, um mesmo serviço pode ser consumido por diferentes sistemas e em diferentes contextos.
- Substituição de Sistemas Legados
  - Como um sistema é formado de vários serviços pequenos e independentes, a migração de um sistema legado pode ser feita de maneira mais natural e gradativa, uma vez que é possível migrar o sistema por “partes” diminuindo o impacto desta mudança drasticamente.



## E as Desvantagens?

- Aqui, temos algumas desvantagens inerentes a sistemas distribuídos:
  - O time de desenvolvimento terá de implementar a comunicação entre os serviços e lidar com possíveis falhas de comunicação;
  - Implementar requisições que são spamadas para vários serviços pode ser uma tarefa complicada, principalmente quando temos mais de um time de desenvolvimento envolvido;
  - Testar interações entre serviços não é uma tarefa das mais fáceis;
  - As ferramentas utilizadas pelo dev team são voltadas a construção de sistemas monolíticos, não oferecendo muitas vezes suporte a construção de sistemas baseados em micro serviços.
- Complexidade de deploy em ambiente de produção:
  - É necessário subir e monitorar diversos micro serviços que possivelmente encontram-se em servidores/VMs diferentes, coordenando o bom funcionamento de todos estes serviços;
- Consumo de memória;



# Quando Utilizar uma Arquitetura de Microservices?

- A aplicação possui um requisito específico de negócio:
  - Por exemplo, uma aplicação que deve armazenar de forma segura dados de cartão de crédito de vários clientes de forma segura e com acesso restrito. Essa aplicação provavelmente precisará escalar de maneira distinta do restante do sistema!!
- Times distintos e separados por serviço:
  - Um dos princípios da arquitetura voltada a micro serviços é que um time pode ser responsável por mais de um micro serviço mas um micro serviço só pode ter um time como responsável.
- Base de dados individual e não compartilhada
  - Quando a aplicação possui uma base de dados que só é acessível via REST ou outro protocolo de comunicação por outros serviços
- Compartilhamento de uma mesma base de dados entre serviços
  - Um serviço específico, sem lógica complexa que só forneça acesso aos dados necessários para outros micro serviços.





## Como definir um Micro Serviço?

- A definição dos micro serviços, ou como a aplicação deve ser desmembrada, é uma decisão complexa e extremamente importante dentro de um projeto orientado a micro serviços;
- Esta divisão deve prezar sempre pela alta coesão e pelo baixo acoplamento entre os serviços;
- Basicamente existem duas abordagens para a definição de um micro serviço:
  - “Capacidades de Negócio”;
  - Subdomínio;



# Como definir um Micro Serviço?

- **Capacidade de Negócio:**
  - Este conceito vem da modelagem de arquitetura de negócios e representa uma atividade que gera valor para o negócio ou um objeto do negócio;
    - Ex: (1) Serviço de Ordem: responsável pelas ordens de serviço do negócio. (2) Serviço de Clientes: responsável por lidar com os clientes;
  - Vantagens:
    - Alta coesão e baixo acoplamento;
    - Times especializados em um domínio específico do negócio;
    - Desde que as capacidades de negócio da organização sejam bem definidas, o microserviço também o será;
  - Desvantagens:
    - Como definir as capacidades de negócio da organização?



# Como definir um Micro Serviço?

- **Domain-Drive Design(DDD):**
  - Este conceito trata o negócio como um domínio e suas várias áreas como subdomínios do problema
  - Vantagens:
    - Alta coesão e baixo acoplamento;
    - Times especializados em um domínio específico do negócio;
    - Desde que as capacidades de negócio da organização sejam bem definidas, o microserviço também o será;
  - Desvantagens:
    - Como definir os subdomínios da organização da organização?



## Como funciona a troca de mensagens em uma aplicação baseada em Micro Serviços?

- Quando utilizamos a arquitetura de micro serviços para desenvolver uma aplicação, é necessário pensar em como estes serviços irão se comunicar com clientes externos e entre si;
- Existem alguns padrões para esta comunicação, sendo:
  - Invocação de Procedimentos Remotos;
  - Mensagens;



## Como funciona a troca de mensagens em uma aplicação baseada em Micro Serviços?

- **Invocação de Procedimentos Remotos(RPI):**
  - Baseado em protocolos request/reply;
    - RPC;
    - REST;
  - Vantagens:
    - Simples e familiar;
    - Fácil de usar;
    - Não necessita de um message broker;
  - Desvantagens:
    - Somente suporta interações baseadas em request/response;
    - Serviços precisam estar ativos durante toda a comunicação;



## Como funciona a troca de mensagens em uma aplicação baseada em Micro Serviços?

- **Mensagens:**
  - Soluções baseadas em protocolos de comunicação de mais baixo nível, como o TCP/IP
    - Apache Kafka;
    - RabittMQ
  - Vantagens:
    - Suportam vários padrões de comunicação;
    - Baixo acoplamento entre os serviços, uma vez que os mesmos não precisam saber previamente onde todos estão;;
    - Tem backup das mensagens(Apache Kafka).
  - Desvantagens:
    - Complexidade.



## E os dados, como são compartilhados?

- Dentro de uma aplicação baseada em micro serviços, o tratamento dos dados é bastante importantes, pois existem diversos serviços que precisam ter acesso aos mesmos dados e é necessário manter a consistência dos dados entre os serviços.
- Quando se trata do tratamento dos dados em arquiteturas de micro serviços existem alguns padrões, sendo os dois a seguir os mais utilizados:
  - Database compartilhada;
  - Uma database por micro serviço.



## E os dados, como são compartilhados?

- **Database compartilhada:**
  - Neste caso, todos os micro serviços operam sob uma única database compartilhada por todos
  - Vantagens:
    - ACID(Atomicity, Consistency, Isolation, Durability);
    - Facilidade;
  - Desvantagens:
    - Acoplamento de desenvolvimento;
    - Acoplamento de execução;
    - Requisitos.





## E os dados, como são compartilhados?

- **Uma Database por Serviço:**

- Neste caso, cada micro serviço opera com a sua própria database replicando as informações necessárias entre as databases;
- Vantagens:
  - Baixo acoplamento;
  - Database modelada de acordo com as necessidades de cada serviço;;
- Desvantagens:
  - Spam dos dados entre as databases;
  - Queries que agregam dados de múltiplas databases;
  - Dificuldade de gerenciamento de bancos SQL e NoSQL(Grafos, documentos, etc.);



## Como funciona a segurança de uma aplicação baseada em micro serviços?

- Quando utilizamos uma arquitetura baseada em micro serviços é necessário garantir que somente um cliente autorizado tenha acesso ao conteúdo de determinado serviço;
- Por isso é necessário a utilização de algum método de autenticação, como o JWT, que autentique o cliente e garanta a sua identidade;
- Nesse caso, utilizando o JWT, é possível anexar o token no body da request de forma a garantir a identidade do cliente;



## Como funciona o Deploy de uma aplicação baseada em Micro Serviços?

- Aqui, temos mais uma decisão crítica de projeto, que pode impactar bastante o sucesso ou não de um projeto baseado na arquitetura de Micro Serviços: como realizar o Deploy da aplicação?
- Existem diversos padrões de deploy, sendo os 4 a seguir os mais utilizados:
  - Múltiplos serviços em um único host;
  - Um serviço por Host;
  - Um serviço por VM;
  - Um serviço por container.



## Como funciona o Deploy de uma aplicação baseada em Micro Serviços?

- **Múltiplos serviços em um único host:**
  - Neste tipo de abordagem, o mesmo host é compartilhado por vários micro serviços;
  - Vantagens:
    - Uso mais eficiente dos recursos;
  - Desvantagens:
    - Conflito de dependências;
    - Conflito de recursos;
    - Não é possível limitar os recursos consumidos por um serviço específico;
    - Monitoramento mais complexo



## Como funciona o Deploy de uma aplicação baseada em Micro Serviços?

- **Um serviço por host**
  - Neste tipo de abordagem, cada serviço tem o seu próprio host;
  - Vantagens:
    - Cada serviço roda de forma isolada dos outros;
    - Não há conflitos de recursos;
    - Não há conflito de dependências;
    - Monitoramento dos serviços se torna menos complexo;;
  - Desvantagens:
    - Utilização menos eficiente de recursos



## Como funciona o Deploy de uma aplicação baseada em Micro Serviços?

- **Um serviço por VM**
  - Neste tipo de abordagem, cada serviço tem sua própria VM dentro de um host que trabalha com virtualização;
  - Vantagens:
    - Cada serviço roda de forma isolada dos outros;
    - Não há conflitos de recursos;
    - Não há conflito de dependências;
    - Monitoramento dos serviços se torna menos complexo;
    - Torna possível limitar os recursos consumidos por um serviço.
  - Desvantagens:
    - Construir e configurar uma VM pode ser bastante custoso



## Como funciona o Deploy de uma aplicação baseada em Micro Serviços?

- Um serviço por Container
  - Neste tipo de abordagem, cada serviço tem sua própria VM dentro de um host que trabalha com virtualização;
  - Vantagens:
    - Cada serviço roda de forma isolada dos outros;
    - Não há conflitos de recursos;
    - Não há conflito de dependências;
    - Monitoramento dos serviços se torna menos complexo;
    - Torna possível limitar os recursos consumidos por um serviço;
    - Extremamente rápido para subir uma nova instância;
  - Desvantagens:
    - Infraestrutura
    - Equipe altamente especializada

## Exemplo

