

## PROGRAMAÇÃO

# Microservices: O que são, e como funcionam?

Por Victor Lima em 17/09/2018

Você já ouviu falar de Microservices? Dez em 1 ou 1 em dez? Essa é a grande pergunta que fazemos quando estamos para decidir se utilizaremos a grande “moda” dos famosos microservices. Logo, antes de chegar a qualquer conclusão vamos entender um pouco melhor o que realmente são microservices ou em português microserviços.

Com o passar dos anos o termo microservices vem se tornando cada vez mais popular no mundo do desenvolvimento web.

A arquitetura de microservices é uma nova forma de organizar as suas aplicações, pois ela propõe que um software seja subdividido em pequenos serviços (micro-sofwares) totalmente independentes um dos outros, que se comunicam entre si para executar todas as funcionalidades da aplicação.

## O que são microservices – em poucas palavras.

Podemos dizer que os microservices são um “novo” estilo ou uma “nova” arquitetura de criação de software. Como o nome sugere, microserviços se referem a um software dividido em pequenos softwares (micro-sofwares) que têm responsabilidades reduzidas e são independentes.

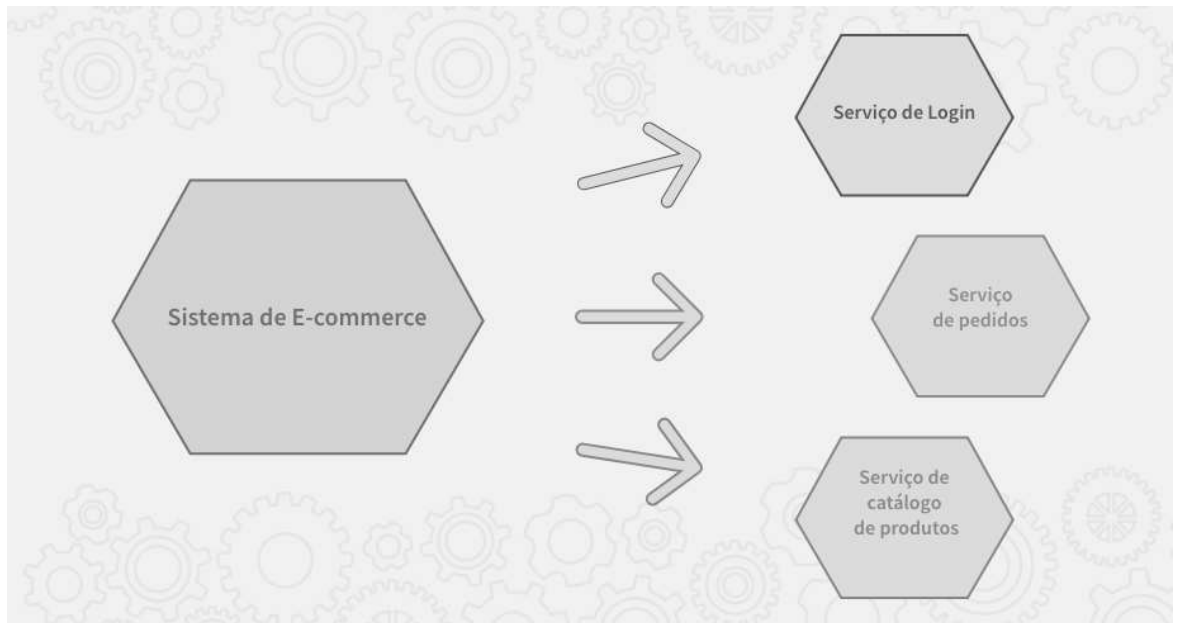
Para exemplificar, imagine que você possui uma loja virtual feita em PHP que implementa diversas funcionalidades, tais como: Cadastro de clientes, gerência de estoque, pedidos, catálogo de produtos, entre outras coisas.

Ou seja, o sistema da loja virtual acaba sendo “inchado” e muito complexo, uma vez que possui muitas funcionalidades e capacidades.



Para uma implementação deste tipo, a arquitetura de microsserviços sugere dividir tal software em diversos softwares independentes (um software para o cadastro de clientes, outro para a gerência de estoque, etc).

Logo, ao invés de haver uma única grande aplicação PHP, é utilizada diversas micro-aplicações, que se comunicam entre si para formar a loja virtual. A imagem abaixo representa muito bem este exemplo.



Em poucas palavras, a arquitetura de microservices permite com que você tenha um único software formado por uma suíte de pequenos serviços. Porém, você deve estar se perguntando, qual é a grande vantagem desta arquitetura em cima das convencionais? Para entender tal vantagem, precisamos primeiramente entender um pouco mais sobre um outro tipo de arquitetura: a monolítica.

## Arquitetura monolítica

Primeiramente, não se assuste com este nome estranho (monolito). A arquitetura monolítica é uma das “mais simples”, e por muitos anos dominou o mundo do desenvolvimento de software. Mas, o que é um software monolítico?

A resposta é simples: um software monolítico é o inverso de um software baseado em microsserviços. O grande conceito por trás da arquitetura monolítica é acoplar todas as funcionalidades de um sistema em um só núcleo.

Imagine o exemplo da loja virtual, onde ao invés de termos micro-sofwares para cada funcionalidade, temos apenas um único software com todas as funcionalidades dentro do mesmo.

Até hoje o desenvolvimento de aplicações monolíticas tomam conta do cenário de desenvolvimento de software, porém empresas que possuem grandes aplicações, como a gigante no mundo de streaming, a Netflix, já está deixando de ter toda sua aplicação em um único arquivo .JAR para utilizar a arquitetura de diversos serviços distribuídos.

## **As vantagens dos microservices sobre os softwares monolíticos.**

Por ser uma abordagem mais simplista e de fácil implementação, uma aplicação de pequeno porte definitivamente deve utilizar o padrão monolítico.

Não há vantagens separar uma aplicação que já é pequena em serviços ainda menores. Nesse caso, algo que deveria ser simples, poderia se tornar muito complexo.

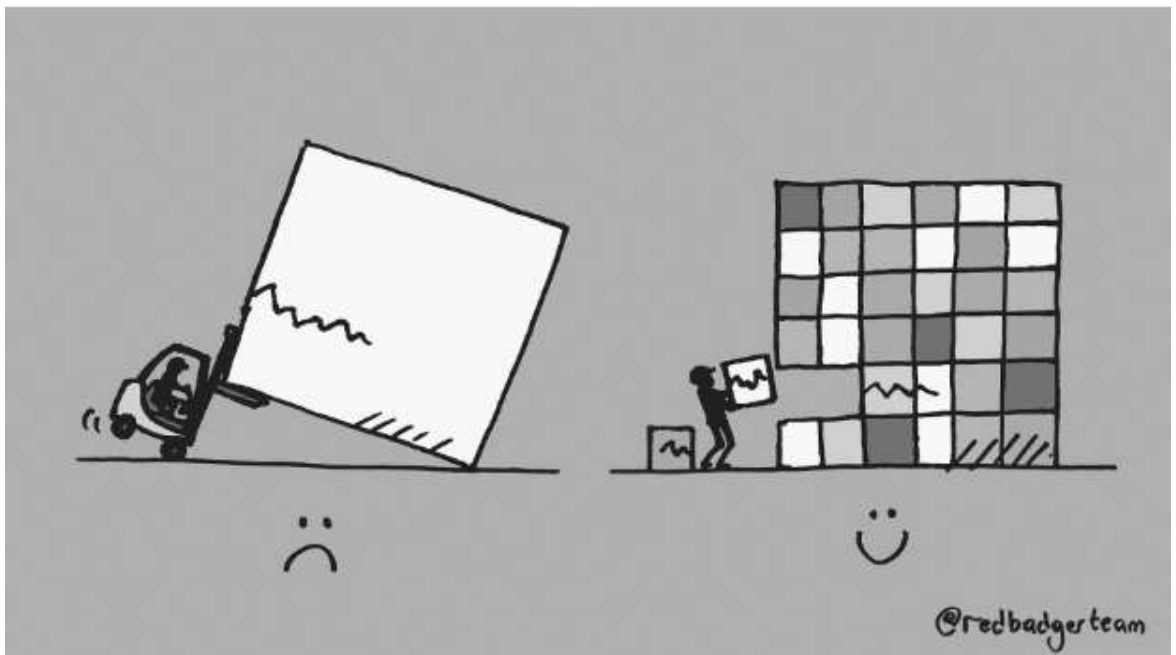
Já para uma implementação grande e/ou sistemas complexos, a arquitetura monolítica acaba deixando de ser eficiente.

Um sistema monolítico grande e complexo, é difícil de manter, difícil de fazer deploy, difícil de escalar, e é extremamente acoplado; isso significa que, qualquer mudança que é realizada em um ponto X de um software, pode afetar um ponto Y fazendo com que todo sistema venha a falhar.

Em um sistema monolítico, cada parte depende da outra. Se uma parte falha, todo o sistema falha; e é baseado nesses problemas que os microsserviços tornam-se vantajosos em relação as arquiteturas monolíticas.

Nos microsserviços, cada serviço é independente, ou seja, ao fazer uma modificação em um serviço X, tal modificação não acaba impactando todo o sistema que utiliza o recurso, uma vez que os serviços são isolados. Isso também implica que, se um serviço falhar, geralmente todo o resto da aplicação fica de pé.

Além disso, o fato de cada parte do sistema ser separado em partes, facilita a escalabilidade do software, pois é muito mais fácil você escalar um projeto que está bem dividido e organizado do que ter que se aventurar dentro de um grande monolito.



## Heterogeneidade de tecnologias

Uma outra grande vantagem de se trabalhar com microservices, é a heterogeneidade de tecnologias. Isso quer dizer que é possível utilizar tecnologias distintas ao mesmo tempo.

Cada serviço pode ser escrito em uma tecnologia diferente, desde que eles consigam se comunicar entre si (se for um projeto web, geralmente esta comunicação é feita via protocolo HTTP através de apis).

Voltando ao exemplo da loja virtual. imagine que queiramos que um recurso de chat fique disponível aos usuários. Ao identificar essa demanda é notado que toda a loja foi desenvolvida em PHP, porém, a equipe de desenvolvimento acredita que a utilização do Node.js é a solução mais adequada para esse tipo de recurso.

Nesse caso, se a arquitetura baseada em microserviços estiver em uso, é totalmente possível criar uma aplicação totalmente escrita com Node.js e tê-la integrada com o sistema principal desenvolvido em PHP.

## Conclusão

Para concluir, dependendo do tamanho e do nível de complexidade dos softwares, é muito mais vantajoso utilizar a arquitetura de microserviços. Esta implementação traz muitos benefícios como: baixo acoplamento entre partes distintas do software, encapsulamento e independência entre os serviços, uma boa tolerância a falhas e uma grande otimização no nível de escalabilidade do projeto.

Porém, esta arquitetura é muito mais complexa de ser implementada. Quando a escolha na utilização de microserviços é tomada, uma análise minuciosa sobre cada elemento e serviço do sistema deve ser feita; além disso, a definição das tecnologias a serem utilizadas, infraestrutura, licenças, processos distintos de deploy também são fundamentais.

De modo geral, a utilização de microserviços veio para ficar no mundo da tecnologia, porém, apesar de todas as vantagens, não podemos nos deixar seduzir e implementar uma arquitetura tão complexa e custosa em projetos de pequeno porte.