

“Microserviços são uma abordagem arquitetônica e organizacional do desenvolvimento de software na qual o software consiste em pequenos serviços independentes que se comunicam usando APIs bem definidas. Esses serviços pertencem a pequenas equipes autossuficientes.” – Amazon

Microserviços, em geral, são aplicações individuais unidas para montar um sistema (mais utilizado em empresas) para expandir a experimentação do programa, e diminuindo a necessidade de criar um projeto enorme, também conhecido como “Monolith”, para atualizar erros em certas partes do programa.

O microserviço contém três princípios:

- **Propósito único:** A liberdade em ser modular permite que certos setores ou companhias em focar a melhorar e reparar certas partes de um programa sem ter a necessidade de atualizar o código inteiro.
- **Baixo Acoplamento:** Em microserviços, cada parte é feita independente da outra, procurando evitar a repetição e reutilização de códigos semelhantes aos outros softwares do sistema. Esta medida evita que se, caso ocorra um bug ou invasão graças à um erro de código em um software, outros não sofram do mesmo problema e comprometam o sistema inteiro.
- **Alta coesão:** cada software obtém a mesma informação do sistema para operar com melhor eficiência comparada a um sistema “Monolith”. Um exemplo sendo Uber, onde o software de localização e o de chamada de motorista são dois softwares diferentes, mas obtém o mesmo set de informação do consumidor para funcionar corretamente.

Com esses princípios, o Microserviços ganha várias vantagens, tal como:

- **Agilidade:** Ao invés de utilizar um software enorme, o pequeno tamanho dos microsserviços diminui o tempo necessário para implementar mudanças.
- **Menores Equipes:** Com menor tamanho de software, os times necessários para manter e atualizar o sistema se torna menor graças a modularidade do tudo.
- **Código Menor:** Com menor tamanho de software, o código necessário acaba diminuindo também, já que não é mais necessário codificar o todo num software só.
- **Escalabilidade:** Com essa modularidade disponível, se permite aumentar e expandir os serviços incluídos no sistema sem ter que se preocupar como que o código interage com o todo. Caso ocorra um erro, é mais fácil de detectar qual outro microsserviço ele está interagindo mal.
- **Isolamento de Dados E Falhas:** A modularidade também permite isolar informações obtidas pelo usuário para permitir a aquisição de padrões e possíveis problemas, além disso, se um código sofrer um erro catastrófico, o programa em todo não será afetado, e o conserto do código se torna mais ágil e simples.

Tudo isso traz uma ótima aparência para microsserviços, mas onde é utilizado?

Uns exemplos são:

- Netflix: Eles exportaram o programa monolith deles para AWS(Amazon Web Services) em 2012, e dividiram em centenas de microsserviços imbutidos no API.
- PayPal
- Amazon
- eBay
- Twitter
- E outros.