



Colégio Técnico de Campinas - Unicamp



Marcelo Morello Manzo
Rafael Moreira Cavalcante de Souza
Vitor Henrique Gírio Paes

Arquitetura de Microsserviços

Campinas - SP
2024
Marcelo Morello Manzo

Rafael Moreira Cavalcante de Souza
Vitor Henrique Gírio Paes

Arquitetura de Microsserviços

Pesquisa acadêmica apresentada ao curso 39 – Desenvolvimento de
Sistemas do Colégio Técnico de Campinas, como requisito para
completar os componentes curriculares da disciplina TI323.
Orientador(a): Simone Pierini Facini Rocha

Campinas - SP
2024

Sumário

1. Resumo.....	4
----------------	---

2. Introdução.....	4
3. Contexto Histórico e Evolução.....	4
4. Importância e Impacto.....	5
5. Definição da Arquitetura de Microsserviços.....	5
6. Exemplo de Arquitetura de Microsserviços	6
7. Exemplo de Implementação de um Microsserviço.....	6
8. Conclusão.....	7
9. Referências.....	8

1. Introdução

Nos últimos anos, as empresas de tecnologia enfrentaram uma pressão crescente para inovar rapidamente e se adaptar às mudanças do mercado. Em resposta a esses desafios, a Arquitetura de Microsserviços surgiu como uma alternativa eficaz às tradicionais arquiteturas monolíticas. Ao contrário delas, em que todos os componentes do sistema estão fortemente acoplados, os microsserviços são projetados para funcionar de forma independente, facilitando a implantação de novos recursos e a manutenção do sistema. Empresas como Amazon e Netflix já adotaram essa abordagem, colhendo os benefícios de uma maior flexibilidade e eficiência no desenvolvimento de software.

2. Contexto Histórico e Evolução

Se olharmos para o passado, veremos que o desenvolvimento de software foi dominado durante muito tempo por sistemas monolíticos. Esses sistemas, embora funcionais, traziam consigo uma série de dificuldades, principalmente em termos de escalabilidade e manutenção. A partir do advento das novas tecnologias de containerização, como o Docker, e de orquestração, como o Kubernetes, que o cenário começou a mudar. Esses avanços tecnológicos abriram as portas para a popularização dos microsserviços, permitindo que as empresas adotassem práticas ágeis e aprimorassem suas operações de DevOps. A partir dessas mudanças, as empresas perceberam que a separação de funções em serviços menores era o caminho para construir software de forma mais eficaz e eficiente.

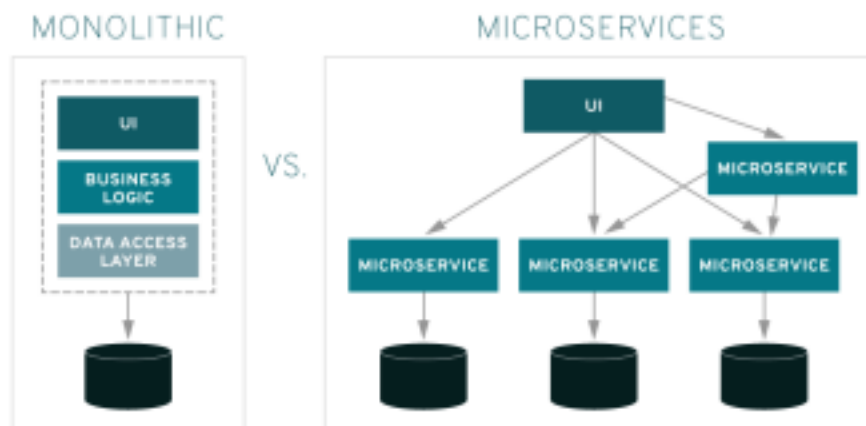


Figura 1 - Sistemas Monolíticos vs Microserviços. Fonte:

<https://www.redhat.com/pt-br/topics/microservices/what-are-microservices>

3. Importância e Impacto

A mudança para uma arquitetura de microserviços não afeta apenas o processo de desenvolvimento de software, mas também transforma a maneira como as organizações operam. Com as equipes trabalhando de forma simultânea e independente em diferentes partes do sistema, o tempo de desenvolvimento reduzirá. Além disso, a arquitetura torna os sistemas mais resilientes pois se um serviço falha, os outros continuam funcionando. Os microserviços estão ajudando as empresas a inovar de forma mais rápida, colaborativa e ágil, permitindo que adotem novas tecnologias conforme necessário.

4. Definição da Arquitetura de Microserviços

Pode-se dizer que a Arquitetura de Microserviços organiza uma aplicação como um conjunto de serviços pequenos e autônomos, cada um focado em uma funcionalidade específica. Cada serviço pode ser implementado e gerenciado de forma independente, o que facilita não apenas a escalabilidade, mas também a manutenção contínua do sistema. Esses serviços comunicam-se entre si por meio de APIs, o que garante a interação eficiente entre as diferentes partes da aplicação. Como resultado, essa abordagem promove uma flexibilidade muito maior, permitindo a utilização de diferentes recursos tecnológicos em cada serviço.

5. Exemplo de Arquitetura de Microsserviços

Um exemplo pode ser observado em plataformas de e-commerce. Imagine que uma loja online se divide em diferentes microsserviços, cada um cuidando de uma parte fundamental da operação: há um serviço responsável pelo catálogo de produtos, outro que gerencia os pedidos e um terceiro que processa pagamentos. Esse modelo permite que cada serviço seja desenvolvido com a linguagem e a tecnologia mais apropriada para sua função específica. A tabela a seguir ilustra estes serviços:

Serviço	Responsabilidade
Serviço de Produto	Gerenciar informações sobre produtos
Serviço de Pedido	Processar e gerenciar pedidos dos usuários
Serviço de Pagamento	Realizar transações financeiras

Figura 3 - Ilustração de diferentes serviços e suas responsabilidades. Fonte: Souza, Rafael Moreira Cavalcante de. (2024).

6. Exemplo de Implementação de um Microsserviço



```
1  const express = require("express");
2  const app = express();
3  const port = 3000;
4  app.use(express.json());
5  app.get("/users", async (req, res) => {
6    users = await databaseUsers.find();
7    res.status(200).json(users);
8  });
9  app.listen(port, () => {
10   console.log(`Server is running on port ${port}`);
11 });
```

A seguir, apresenta-se um exemplo de código que ilustra a estrutura básica de um microserviço em Node.js utilizando o framework Express:

Figura 2 - Estrutura de um microserviço em Node.js.. Fonte: Souza, Rafael Moreira Cavalcante de. (2024).

Este microserviço é responsável por listar todos os usuários de um sistema. Cada serviço pode ser desenvolvido com uma linguagem diferente ou em uma arquitetura distinta, desde que siga o princípio da comunicação via API.

7. Conclusão

A Arquitetura de Microserviços trouxe uma mudança sobre como o software é projetado, permitindo que as organizações respondam rapidamente às exigências do mercado, com maior resiliência e escalabilidade, mas também apresenta desafios significativos.

Vantagens:

- **Escalabilidade:** Cada serviço pode ser ajustado conforme a demanda, permitindo maior flexibilidade.
- **Resiliência:** A falha de um serviço não compromete o sistema como um todo.
- **Agilidade:** Equipes podem desenvolver funcionalidades de forma

independente, acelerando entregas.

Desvantagens:

- **Complexidade:** A gestão de múltiplos serviços exige ferramentas especializadas.
- **Comunicação:** A interação entre serviços pode gerar latência e novos pontos de falha.
- **Segurança:** A exposição de múltiplas APIs aumenta a superfície de ataque, exigindo maior controle.

Sendo assim, a arquitetura de microsserviços é uma ferramenta que demanda planejamento estratégico e preparo técnico, garantindo que suas vantagens superem os desafios. É fundamental analisar as necessidades do projeto para assegurar que essa abordagem seja implementada da forma correta.

8. Referências Bibliográficas

GOOGLE CLOUD. O que é arquitetura de microsserviços. 2024. Disponível em: [https://cloud.google.com/learn/what-is-microservices-architecture?hl=pt-BR#:~:text=A%20arquitetura%20de%20microsservi%C3%A7os%20\(geralmente,parte%20tendo%20sua%20pr%C3%B3pria%20responsabilidade](https://cloud.google.com/learn/what-is-microservices-architecture?hl=pt-BR#:~:text=A%20arquitetura%20de%20microsservi%C3%A7os%20(geralmente,parte%20tendo%20sua%20pr%C3%B3pria%20responsabilidade). Acesso em: 16 out. 2024.

RED HAT. O que são microsserviços?. 2024. Disponível em: <https://www.redhat.com/pt-br/topics/microservices/what-are-microservices>. Acesso em: 16 out. 2024.

XP INC. Entendendo a arquitetura de microsserviços. 2024. Disponível em: <https://medium.com/xp-inc/entendendo-a-arquitetura-de-microservices-cdab6b52d6ed>. Acesso em: 17 out. 2024.

AWS AMAZON. O que são microsserviços?. 2024. Disponível em: <https://aws.amazon.com/pt/microservices/> em: 17 out. 2024.