

**MINISTÉRIO DA EDUCAÇÃO**

**UNIVERSIDADE FEDERAL RURAL DA AMAZÔNIA**

**CAMPUS CAPITÃO POÇO**

**BACHARELADO EM SISTEMAS DE INFORMAÇÃO**

**DESENVOLVIMENTO WEB**

**Anthony Ivens da Silva Ferreira**

**Capitão poço – PA**

**2024**

**Nap01Parte01**

**Link Github:** [**https://github.com/ivensanthony/produto-api.git**](https://github.com/ivensanthony/produto-api.git)

**Link vídeo: https://drive.google.com/drive/folders/1rALH1vKEPp\_J2EiaJq2hTKsfhj28Ky3f**

**Nap01Parte02**

**ARQUITETURA DE MICROSSERVIÇOS**

**1 INTRODUÇÃO**

A arquitetura de desenvolvimento web tem evoluído consideravelmente nas últimas décadas, buscando atender às crescentes demandas por escalabilidade, manutenção e resiliência em sistemas modernos. Neste contexto, a Arquitetura de Microsserviços tem ganhado destaque por permitir a construção de aplicações compostas por serviços independentes e altamente coesos.

**2 CONCEITO**

A Arquitetura de Microsserviços é um estilo arquitetural onde uma aplicação é composta por pequenos serviços autônomos que comunicam-se entre si através de protocolos leves, como o HTTP ou mensagens assíncronas (mensageria). Cada serviço é desenvolvido, implantado e escalado de forma independente, sendo responsável por uma funcionalidade ou domínio de negócio específico (NEWMAN, 2015).

**3 CARACTERÍSTICAS PRINCIPAIS**

Dentre as principais características da arquitetura de microsserviços, destacam-se:

- Descentralização: cada serviço possui seu próprio ciclo de vida, podendo ser mantido e atualizado sem interferir nos demais.

- Independência tecnológica: diferentes serviços podem ser escritos em linguagens distintas ou utilizar bancos de dados específicos.

- Escalabilidade independente: é possível escalar apenas os serviços que exigem maior capacidade, otimizando recursos.

- Implantação contínua: favorece a integração e entrega contínuas (CI/CD), aumentando a agilidade nas atualizações.

**4 VANTAGENS**

A adoção da arquitetura de microsserviços oferece diversas vantagens:

- Facilidade de manutenção: o código é mais organizado e segmentado, facilitando a identificação e correção de erros.

- Resiliência: falhas em um serviço não comprometem o sistema como um todo, aumentando a tolerância a falhas.

- Agilidade no desenvolvimento: times podem trabalhar de forma paralela em diferentes serviços.

- Adoção progressiva de novas tecnologias: serviços isolados permitem testes com novas ferramentas ou linguagens sem impactar o sistema completo.

**5 DESVANTAGENS**

Apesar das vantagens, essa arquitetura também apresenta desafios:

- Complexidade na comunicação: a interdependência entre serviços exige um controle eficiente da comunicação, que pode gerar sobrecarga.

- Gerenciamento distribuído: monitorar, testar e versionar diversos serviços exige ferramentas e práticas específicas.

- Overhead de implantação: configurar e manter pipelines de CI/CD para múltiplos serviços pode demandar tempo e recursos.

**6 LINGUAGENS DE PROGRAMAÇÃO ASSOCIADAS**

A arquitetura de microsserviços não se restringe a uma linguagem específica. A seguir, são apresentadas duas linguagens amplamente utilizadas neste contexto:

**6.1 Java**

A linguagem Java, em conjunto com o framework Spring Boot, é uma das mais utilizadas para a construção de microsserviços. O suporte à injeção de dependência, controle de versões, e integração com ferramentas como Spring Cloud e Netflix OSS tornam o ecossistema Java robusto para esse tipo de arquitetura.

Exemplo: um microsserviço desenvolvido com Spring Boot pode expor um endpoint REST para gerenciar pedidos em um sistema de e-commerce, utilizando um banco de dados PostgreSQL via JPA.

**6.2 JavaScript (Node.js)**

O Node.js, baseado em JavaScript, é outra escolha popular. Ele permite a criação de serviços leves e rápidos, ideais para arquiteturas assíncronas e com alta demanda de I/O. Frameworks como Express.js facilitam a criação de APIs RESTful.

Exemplo: um microsserviço com Node.js pode ser responsável pelo envio de notificações por e-mail, sendo escalado conforme a carga de envio.

**7 CONSIDERAÇÕES FINAIS**

A arquitetura de microsserviços representa uma abordagem moderna e flexível para o desenvolvimento de aplicações web complexas. Ao adotar este estilo arquitetural, as equipes podem ganhar em agilidade, resiliência e escalabilidade, embora seja necessário enfrentar uma maior complexidade na infraestrutura e no gerenciamento.

**REFERÊNCIAS**

NEWMAN, Sam. Building Microservices: Designing Fine-Grained Systems. O'Reilly Media, 2015.

FOWLER, Martin; LEWIS, James. Microservices: a definition of this new architectural term. 2014. Disponível em: <https://martinfowler.com/articles/microservices.html>. Acesso em: 18 jul. 2025.