

Princípios Arquiteturais REST e Arquitetura de Microserviços: Integração e Benefícios

João Pedro R. Leite¹, Guilherme B. Spiger²

¹Tecnologia em Sistemas para Internet
Universidade Tecnológica Federal do Paraná (UTFPR)
Rua Cristo Rei, 19 - Vila Becker - CEP 85902-490 - Toledo-PR

joaopedroleite@alunos.utfpr.edu.br

Abstract. *In this article, we'll explore how REST Architectural Principles and Microservices Architecture complement each other to create more robust and adaptable systems. We'll break down the core concepts behind REST and Microservices Architecture, and see how these approaches can provide significant benefits. We'll also discuss the challenges that arise when implementing these architectures and key considerations to keep in mind.*

Resumo. *Neste artigo, vamos explorar como os Princípios Arquiteturais REST e a Arquitetura de Microserviços se complementam para criar sistemas mais robustos e adaptáveis. Vamos detalhar os conceitos essenciais por trás do REST e da Arquitetura de Microserviços, e ver como a junção dessas abordagens pode trazer grandes vantagens. Também discutiremos os desafios que surgem ao implementar essas arquiteturas e as principais considerações a ter em mente.*

1. Introdução

A evolução das demandas por sistemas distribuídos, escaláveis e resilientes impulsionou o desenvolvimento de novas abordagens arquiteturais, como os Princípios Arquiteturais REST e a Arquitetura de Microserviços. Ambas as arquiteturas, embora distintas em seu foco, compartilham a busca por flexibilidade e eficiência no desenvolvimento de sistemas modernos.

Os Microserviços incentivam a divisão de sistemas complexos em serviços independentes que se comunicam por meio de APIs. Já o REST, como estilo arquitetural, define princípios que regem a comunicação na web de forma distribuída, escalável e flexível, como descrito por Júlia Leôncio Rodrigues e Giuliano Scombatti Pinto [1].

Este trabalho tem como objetivo explorar a complementaridade entre essas abordagens, seus benefícios e os desafios encontrados na integração desses dois paradigmas, com foco na aplicação em ambientes de desenvolvimento distribuído.

2. Princípios Arquiteturais REST

Os Princípios REST (Representational State Transfer), introduzidos por Roy Fielding, propõem uma abordagem arquitetural para a construção de sistemas distribuídos, baseados nos princípios fundamentais da web [2]. REST está fortemente associado ao protocolo HTTP e promove um estilo de comunicação stateless, em que cada interação cliente-servidor deve ser independente, melhorando assim a escalabilidade do sistema [1].

2.1. Princípios Fundamentais do REST

Os princípios fundamentais do REST são:

- **Stateless:** Cada requisição do cliente ao servidor deve conter todas as informações necessárias para seu processamento. Isso elimina a necessidade de armazenamento de estado no servidor, permitindo que os sistemas REST escalem de forma eficiente.
- **Interface Uniforme:** REST promove o uso de uma interface uniforme, garantindo que todos os recursos sejam acessíveis por uma URI e manipulados com métodos HTTP bem definidos (GET, POST, PUT, DELETE) [2].
- **Arquitetura Cliente-Servidor:** O REST mantém a separação clara entre cliente e servidor, o que facilita a evolução de ambos de maneira independente [1].
- **Cacheability:** As respostas HTTP em sistemas REST podem ser armazenadas em cache para aumentar a eficiência e reduzir a latência das interações.

2.2. Vantagens e Desvantagens do REST

Vantagens:

- **Escalabilidade:** A natureza stateless do REST e seu suporte ao cache permitem que os sistemas sejam altamente escaláveis, adequados para lidar com grandes volumes de tráfego.
- **Interoperabilidade:** REST utiliza tecnologias amplamente adotadas, como HTTP e JSON, tornando-o ideal para a integração com diversos sistemas e plataformas [1].

Desvantagens:

- **Segurança:** A exposição de APIs REST pode aumentar a superfície de ataque de um sistema, exigindo robustas medidas de segurança, como autenticação e autorização adequadas [4].
- **Overhead de Rede:** Como REST depende do HTTP, pode haver overhead significativo ao lidar com interações frequentes e grandes volumes de dados.

3. Arquitetura de Microserviços

Na Arquitetura de Microserviços, grandes aplicações são fragmentadas em pequenos serviços autônomos, cada um responsável por uma funcionalidade específica do sistema. Cada microserviço pode ser desenvolvido, testado, implantado e escalado de forma independente [3]. O uso dessa arquitetura se torna cada vez mais comum, com grandes empresas como Amazon e Netflix adotando essa abordagem para melhorar a escalabilidade e resiliência de seus sistemas [1].

3.1. Características dos Microserviços

Entre as principais características dos microserviços estão:

- **Desacoplamento:** A independência entre serviços permite que os mesmos sejam desenvolvidos, atualizados e escalados de maneira isolada, promovendo maior flexibilidade no desenvolvimento [4].
- **Escalabilidade Independente:** Cada serviço pode ser escalado de acordo com suas necessidades específicas, otimizando o uso de recursos do sistema.

4. Integração entre REST e Microserviços

A combinação de REST com Microserviços permite uma abordagem modular e eficiente, ideal para ambientes de sistemas distribuídos. O uso de APIs RESTful como interface para comunicação entre microserviços é amplamente adotado devido à simplicidade e interoperabilidade oferecidas [1, 4].

4.1. Benefícios da Integração

Interface Uniforme : O uso de APIs REST entre microserviços proporciona uma interface clara e padronizada para a comunicação entre diferentes componentes do sistema.

Escalabilidade e Desempenho : Microserviços com comunicação baseada em REST podem ser escalados independentemente, enquanto o uso de cache e a arquitetura stateless garantem desempenho eficiente [1, 4].

5. Conclusão

Os Princípios REST, quando aplicados a uma Arquitetura de Microserviços, fornecem uma base robusta para sistemas distribuídos modernos. Embora ambas as abordagens apresentem desafios, como a gestão de segurança e complexidade operacional, os benefícios em termos de escalabilidade, resiliência e flexibilidade superam as dificuldades, permitindo o desenvolvimento de sistemas adaptáveis e resilientes [1, 4].

Referências

- [1] RODRIGUES, J. L.; PINTO, G. S. *Análise da Arquitetura de MicroServiços*. FATEC, 2018.
- [2] FIELDING, R. T. *Architectural Styles and the Design of Network-based Software Architectures*. University of California, 2000.
- [3] NEWMAN, S. *Building Microservices: Designing Fine-Grained Systems*. O'Reilly Media, 2015.
- [4] AMARAL, O.; CARVALHO, M. *Arquitetura de Micro Serviços: uma Comparação com Sistemas Monolíticos*. UFPB, 2017.