

Roteiro de estudos para avançar na Programação: Sugestões iniciais voltadas a Microserviços

Pedro Antonio Carlos Bastos, Sérgio Santos Silva Filho, Darley Domingos de Almeida, Toni Amorim de Oliveira, Max Robert Marinho

Ciência da Computação – Universidade Estadual do Mato Grosso (UNEMAT)
Rua Santa Rita, 128 – 78.780-000 – Alto Araguaia – MT – Brasil

pedro.bastos@unemat.br, santos@unemat.br, darley.domingos@unemat.br,
toniamorim@unemat.br, max.marinho@unemat.br

Abstract. *This article presents a study roadmap aimed at assisting students and future professionals in the programming career. Based on preliminary research, we seek to develop a guide grounded in bibliographic references and the analysis of roadmaps presented by Roadmap.sh, applying these guidelines for educational purposes with a focus on microservices. The study addresses the essential concepts for learning programming, highlighting tools and recommended practices that align with guidelines from organizations such as ACM (Association for Computing Machinery) and IEEE (Institute of Electrical and Electronics Engineers). We propose a simple yet potentially effective roadmap in the teaching-learning process, emphasizing the relevance of the programming field in the current context.*

Resumo. *Este artigo apresenta uma proposta de roteiro de estudos destinada a auxiliar estudantes e futuros profissionais na carreira de programação. Com base em pesquisas iniciais, buscamos desenvolver um guia fundamentado em referências bibliográficas e na análise de roteiros apresentados pelo Roadmap.sh, aplicando essas diretrizes para fins didáticos, com foco em microserviços. O estudo aborda os principais conceitos essenciais para o aprendizado de programação, destacando ferramentas e práticas recomendadas que seguem diretrizes de órgãos como a ACM (Association for Computing Machinery) e a IEEE (Institute of Electrical and Electronics Engineers). Propomos um roteiro simples, mas potencialmente eficaz, no processo de ensino-aprendizagem, enfatizando a relevância da área de programação no contexto atual.*

1. Introdução

Microserviços são uma abordagem arquitetural que organiza uma aplicação como um conjunto de serviços pequenos, independentes e autônomos, que se comunicam entre si por meio de APIs. Essa arquitetura permite que cada serviço seja desenvolvido, implantado e escalado de forma independente, facilitando a manutenção e a evolução da aplicação como um todo (Newman, 2015).

Lewis e Fowler (2014) destacam as seguintes características principais: 1. Independência: cada microserviço pode ser desenvolvido e implantado de maneira autônoma; 2. Escalabilidade: possibilita que serviços específicos sejam escalados conforme necessário, sem afetar outros serviços; 3. Tecnologia Agnóstica: cada microserviço pode ser implementado em uma linguagem ou framework diferente, de acordo com a necessidade; e 4. Resiliência: a falha de um microserviço não compromete necessariamente a operação dos demais.

Um exemplo claro dessa abordagem é o uso da linguagem Java, amplamente adotada na construção de microserviços devido à sua robustez, desempenho e suporte a diversas

bibliotecas e frameworks. Um dos frameworks mais populares para a criação de microserviços em Java é o Spring Boot, que simplifica o desenvolvimento de aplicações autônomas baseadas no ecossistema Spring.

Nesse sentido, a estrutura proposta por Newman (2015) consiste em etapas básicas para o desenvolvimento de microserviços: 1. Criação do Projeto: geração de um esqueleto de projeto com as dependências necessárias, como Spring Web e Spring Data JPA; 2. Configuração do Servidor: o Spring Boot inclui um servidor embutido (como Tomcat), facilitando o processo de desenvolvimento e testes; e 3. Definição de REST APIs, que permitem a comunicação entre microserviços. O Spring MVC, por sua vez, facilita a criação de controladores que manipulam requisições HTTP. Posteriormente, é essencial abordar a Comunicação entre Microserviços, a Persistência de Dados e, finalmente, a Resiliência e Escalabilidade.

Neste trabalho, apresentamos uma proposta de roteiro baseada em referências bibliográficas recomendadas pelas diretrizes curriculares, tanto da ACM quanto da IEEE, bem como também, nas diretrizes do Roadmap.sh¹. O objetivo é desenvolver um roteiro de estudos destinado a auxiliar estudantes e futuros profissionais na carreira de programação, aplicando diretrizes para fins didáticos, com foco em microserviços. Espera-se que, ao ser aplicado, esse roteiro sirva como um recurso valioso para auxiliar futuros programadores em seu aprendizado na linguagem Java, mesmo que de maneira básica e introdutória.

2. Proposta, Materiais e Método

Para o desenvolvimento desta proposta, o roteiro é estruturado em etapas que cobrem desde os conceitos fundamentais de microserviços até a implementação prática com Java, utilizando o framework Spring Boot. A primeira etapa será a introdução dos conceitos teóricos, onde os alunos terão contato com a arquitetura de microserviços, suas características principais, e as vantagens em relação às arquiteturas monolíticas. Serão discutidos também aspectos como resiliência, escalabilidade e a independência dos serviços, com foco na aplicação prática dessas teorias em projetos reais.

Na segunda etapa, o guia irá orientar os estudantes na criação e configuração de um ambiente de desenvolvimento apropriado para a construção de microserviços em Java. Aqui, serão abordadas ferramentas essenciais, como o Maven para gerenciamento de dependências, e o Docker, para a criação de containers que facilitam a execução isolada de microserviços. Além disso, os alunos aprenderão sobre a configuração de servidores embutidos no Spring Boot e o uso de APIs REST para intercomunicação entre os serviços, com exemplos práticos para facilitar o entendimento.

Por fim, a última etapa do roteiro será voltada à implementação de funcionalidades avançadas, como a comunicação assíncrona entre microserviços e a persistência de dados utilizando bancos de dados relacionais e não relacionais. Além disso, será dado enfoque à escalabilidade das soluções e ao uso de práticas recomendadas para garantir a resiliência dos sistemas. Espera-se que, ao concluir o roteiro, os alunos estejam aptos a desenvolver soluções baseadas em microserviços e compreendam como aplicar esses conhecimentos

¹ **Roadmap.sh:** Web site que disponibiliza roteiros, melhores práticas, projetos, artigos, recursos e jornadas criadas pela comunidade para ajudar na escolha de seu caminho e crescimento em sua carreira.

em contextos mais amplos da programação, contribuindo para o seu crescimento profissional.

3. Considerações Finais e Trabalhos Futuros

Com base nos estudos realizados e na proposta de roteiro apresentada, esperamos desenvolver um método semiótico, baseado em Filho (2016), que possa ser adotado e integrado em um plano pedagógico, visando ser aplicado a uma turma de alunos interessados em aprender programação. Nossa intenção é realizar testes que nos permitam avaliar se os resultados indicam uma possível melhoria no aprendizado e um aumento do interesse dos alunos pela área da programação. Embora o estudo esteja em sua fase inicial, acreditamos que, após análises mais aprofundadas e ações concretas, seremos capazes de alcançar resultados promissores. Para trabalhos futuros, prevemos investigações que explorem outras áreas da programação, como o desenvolvimento web, que está em alta atualmente (ANDREESSEN, 2011; ZELDMAN, 2011; GOLDSMITH, 2017; e EICH, 2014)

Referências

- ACM (Association for Computing Machinery) (2013). “Computing curricula: 2013 curriculum guidelines for undergraduate degree programs in computer science.” Disponível em: <https://www.acm.org/binaries/content/assets/education/cs2013.pdf>.
- Andreesen, M. (2011) “Why software is eating the world.” The Wall Street Journal. Disponível em: <https://www.wsj.com/articles/SB1000142405311190348090457651225091562942302>.
- Eich, B. (2014) JavaScript: The Definitive Guide. 7. ed. Sebastopol, CA: O'Reilly Media.
- Filho, S. S. (2016). Uma abordagem semiótica para o desenvolvimento de WebQuests. *Dissertação (Programa de Mestrado em Ciência da Computação) - Faculdade Campo Limpo Paulista - FACCAMP*, Campo Limpo Paulista, SP. Disponível em: <https://www.cc.faccamp.br/Dissertacoes/SergioSantos.pdf>.
- Goldsmith, K. (2017) “What is web development?” Disponível em: <https://www.oreilly.com/library/view/what-is-web/9781491978783/>. Acesso em: 18 set. 2024.
- IEEE (Institute of Electrical and Electronics Engineers). IEEE Computer Society curriculum guidelines for undergraduate degree programs in computer engineering. 2016. Disponível em: <https://www.computer.org/education/department/ccg>. Acesso em: 15 set. 2024.
- Lewis, J.; Fowler, M. Microservices. Disponível em: <https://martinfowler.com/articles/microservices.html> >. Acesso em: 21 set. 2024.
- Newman, S. (2015) Building microservices: designing fine-grained systems. 1. ed. Sebastopol, CA: O'Reilly Media.
- Norvig, P.; Thrun, S. (2010) “Artificial intelligence: a modern approach.” 3. ed. Upper Saddle River, NJ: Prentice Hall.
- Zeldman, J. (2011) Designing with web standards. 3. ed. Berkeley: New Riders.