



[Página principal do livro](#)

Compre na [Amazon](#) ou [UmLivro](#)

Engenharia de Software Moderna

Prefácio

“

A inutilidade dos prefácios é um lugar comum da história dos prefácios, portanto serei breve. – Eduardo Giannetti

A ideia de escrever este livro surgiu no início de 2019, quando fui alocado para ministrar a disciplina Engenharia de Software, do Bacharelado em Ciência da Computação, da UFMG. Para preparar o curso, comecei com uma análise dos principais livros de Engenharia de Software. Para minha surpresa, percebi que eles tinham mudado pouco desde que cursei a disciplina na minha graduação há mais de 25 anos!

Meu objetivo era escolher um livro que permitisse, no início de uma aula, dizer para os alunos: “hoje vamos estudar tal assunto, que corresponde a tal capítulo do livro-texto”. No final da aula, gostaria de sugerir aos alunos: “para fixar a matéria que acabamos de ver, sugiro que façam tais exercícios”. No entanto, infelizmente, não encontrei esse livro. Em vez disso, tive que fazer uma extensa pesquisa e leitura de pelo menos 15 livros.

Como resultado, preparei mais de 600 slides, que considero conter o principal material que deve ser tratado em uma disciplina de graduação em Engenharia de Software, especificamente em cursos que possuem uma única disciplina na área. Porém, estudar apenas por slides não proporciona a mesma experiência de aprendizado obtida com a leitura atenta de um texto completo e contextualizado.

Assim, surgiu a ideia de transformar os slides em um livro que pudesse desempenhar o papel do sonhado livro-texto, já visando a futuras ofertas da disciplina. E que fosse útil também para outros professores, que devem enfrentar problemas semelhantes ao meu quando têm que ministrar um curso de graduação em Engenharia de Software.

Gostaria, então, de destacar os seguintes pontos sobre o conteúdo, a organização e a estratégia usada na escrita do livro:

- Ele foi escrito para ser um livro moderno, com ênfase em técnicas e princípios que são largamente usados na construção de software nos dias de hoje.
- Por outro lado, o livro também cobre técnicas e princípios tradicionais, porém de forma rápida. O motivo é que achamos importante cobrir a história da área, mencionando o que deu certo e o que não deu. Achamos que essa visão é importante na formação e amadurecimento dos alunos.
- O livro inclui inúmeros exemplos e discussões de casos reais. Para isso, criei a seção “Mundo Real”, na qual os assuntos são ilustrados com exemplos recentes e reais, provenientes de grandes empresas de software e também de artigos científicos, buscando-se apresentar o que se faz de melhor tanto na indústria como na academia.
- O livro foi escrito em português, pois o objetivo é contribuir, primeiro, com os cursos e alunos brasileiros. Porém, optamos por não traduzir alguns termos — como *refactoring*, *branches* e *sprint* — pois achamos que eles são usados, também sem tradução, pelos desenvolvedores brasileiros no seu dia a dia.
- Apesar de tratar de temas modernos, a intenção foi escrever um livro duradouro. Por isso, temas e tecnologias que ainda não passaram pelo teste do tempo não são abordados (ou são abordados de forma rápida). Um exemplo são os últimos frameworks e arquiteturas para implementação de sistemas, que tendem a mudar rapidamente. Também não acoplamos o livro a nenhuma linguagem de programação. Por exemplo, os trechos de código são mostrados em uma sintaxe neutra.
- Ele não é um livro extenso, com diversos capítulos que, na prática, tratam de assuntos que não são importantes nos cursos atuais.
- Cada vez mais, engenheiros de software têm que escrever código. Hoje, há pouco espaço para dizer que eu não preciso programar, pois sou arquiteto ou analista. Por isso, nos capítulos de projeto, testes e refatoração procuramos seguir a recomendação de Linus Torvalds (criador do Linux): “falar é fácil, mas mostre-me o código”. Assim, esses capítulos incluem dezenas de exemplos de código, que simulam problemas e soluções típicos de sistemas reais.
- Ao longo da escrita, procuramos praticar o que enfatizamos no livro, principalmente no que diz respeito à prevalência atual de métodos de desenvolvimento ágeis. Assim, o livro foi escrito seguindo princípios ágeis. Cada capítulo foi tratado como sendo um *sprint*; uma vez pronto, ele foi disponibilizado para uso, para receber críticas e sugestões. Como ocorre com software, acho arriscado, nos dias de hoje, passar anos escrevendo um manuscrito, trancado em minha sala, para só então torná-lo público.

Público-Alvo

O livro destina-se a alunos de cursos de graduação. Ele foi escrito para ser adotado em cursos que possuem uma única disciplina de Engenharia de Software, com 60 horas. Porém, achamos

também que ele pode ser usado — junto com outros livros — em cursos com duas ou mais disciplinas na área.

Além disso, ele pode ser adotado em cursos técnicos e tecnológicos. Na verdade, fizemos um esforço para usar uma linguagem clara, próxima à linguagem coloquial, exatamente para não criar barreiras à adoção do livro. Por fim, também escrevemos o livro pensando em profissionais da área, que estão em busca de aperfeiçoamento em temas e métodos modernos de Engenharia de Software.

Pré-requisitos

Os leitores devem ter domínio de conceitos básicos de programação e de algoritmos e estruturas de dados. Além disso, recomendamos domínio de orientação a objetos. Supondo um curso de graduação de 4 anos, os alunos devem estar na metade final para melhor aproveitar o conteúdo do livro.

Website

O livro possui uma versão aberta, em HTML, disponível em:

<https://engsoftmoderna.info>

Nesse site, estão também disponíveis diversos exercícios de múltipla escolha. Não menos importante: para reportar qualquer erro, mesmo que um simples erro ortográfico, você pode usar este [formulário](#).

Marco Tulio Valente

Belo Horizonte, 10 de fevereiro de 2020.

Sobre o Autor

Marco Tulio Valente é doutor em Ciência da Computação pela Universidade Federal de Minas Gerais, onde atualmente é Professor Associado do Departamento de Ciência da Computação. Seus interesses de pesquisa concentram-se em Engenharia de Software, especificamente nas áreas de Manutenção e Evolução de Software, Qualidade de Software e Mineração de Repositórios de Software. Ele é Bolsista de Produtividade Nível 1D do CNPq e bolsista do Programa Pesquisador Mineiro da FAPEMIG. É autor de mais de 100 artigos em periódicos e conferências internacionais, muitos deles em co-autoria com pesquisadores internacionais de renome. Orientou mais de 30 dissertações de mestrado e cerca de uma dezena de teses de doutorado. Desde 2010, coordena o Applied Software Engineering Research Group ([ASERG](#)), do DCC/UFMG. É responsável também pelo [CSIndexbr](#), sistema que oferece dados transparentes sobre a produção científica brasileira em Computação.



Agradecimentos

Gostaria de agradecer aos mais de 30 alunos que passaram pelo grupo de pesquisa ASERG na última década. Parte relevante do conteúdo deste livro tem sua origem nos trabalhos do nosso grupo. Gostaria de agradecer aos professores Eduardo Figueiredo e André Hora, colegas de departamento, que revisaram alguns capítulos do livro. E ao Prof. José Luis Braga, que fez uma leitura cuidadosa e deu sugestões valiosas para melhoria do texto. Gostaria de agradecer aos alunos do curso de pós-graduação que ofertamos no segundo semestre de 2019 e que ajudaram a calibrar e refinar o conteúdo deste livro. Especialmente, a Ulisses Fernandes e Fábio Ferreira, pela cuidadosa revisão. Gostaria de agradecer a Aline Brito, que ajudou na conversão do texto para o formato final, e a João Eduardo Montandon e Hudson Borges, pela revisão do apêndice sobre Git. Gostaria de agradecer ainda a Osvaldo e Cynthia Valente, pela revisão gramatical de partes do livro.

Modo noturno

Direitos autorais reservados. Versão para uso pessoal. Para reportar quaisquer erros, incluindo pequenos erros de ortografia, use este [formulário](#).