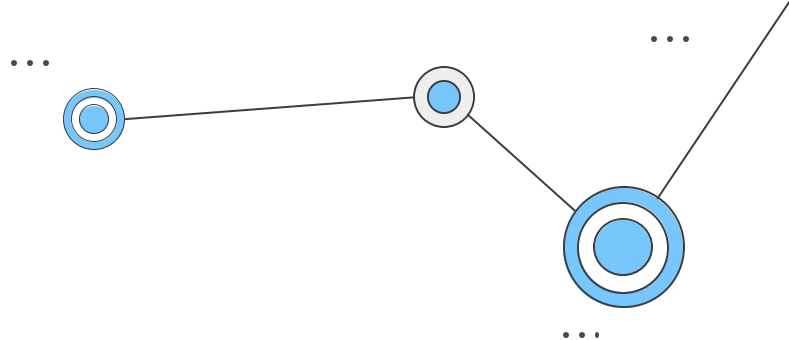




PUC Minas



Projeto de Software

Prof. Dr. João Paulo Aramuni



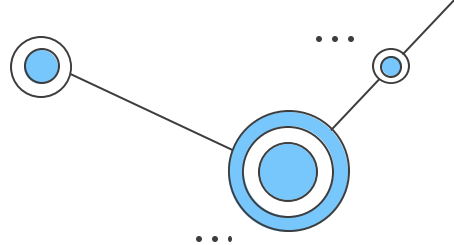
Unidade 1

Introdução

PDS - Manhã /Noite



Introdução a Projeto de Software



Princípios básicos de projetos

Divisão de responsabilidades

- Divida seu aplicativo em recursos distintos com o pouca sobreposição na funcionalidade.

Princípio da responsabilidade única

- Cada componente ou módulo deve ser responsável para apenas um recurso ou uma funcionalidade específica ou agregação de funcionalidade coesa.

Princípio do Conhecimento Mínimo (também conhecido como Lei de Demeter)

- Um componente ou objeto não deve saber sobre detalhes internos de outros componentes ou objetos.

Não se repita (Don't repeat yourself)

- Uma funcionalidade em um único local.

Minimize o design inicial.

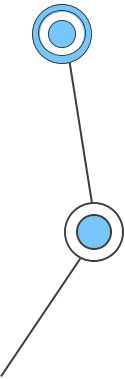
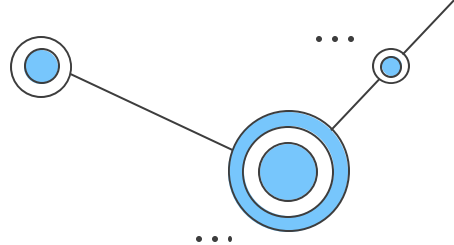
- Apenas projete o que for necessário.



Introdução a Projeto de Software

Boas práticas em projetos

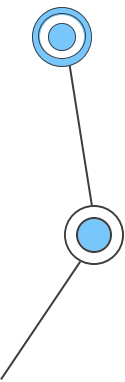
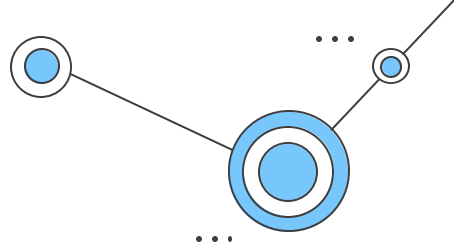
- Mantenha padrões de projeto consistentes dentro de uma mesma camada.
 - Mantenha o design de componentes consistente para uma determinada operação dentro de uma camada lógica.
 - Avaliar quando a complexidade e variação dos requisitos é muito alta.



Introdução a Projeto de Software

Boas práticas em projetos

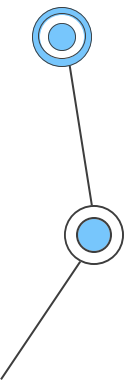
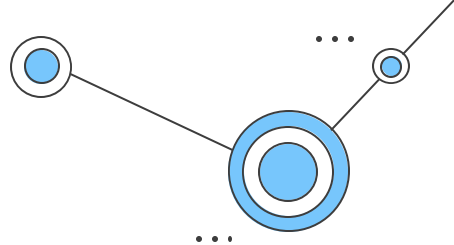
- Não duplique funcionalidades.
 - Aumenta a coesão.
 - Facilita a otimização e manutenção.
- Estabeleça um estilo de codificação e uma convenção de nomenclatura.
 - Facilita o entendimento.
 - Otimiza a manutenção.



Introdução a Projeto de Software

Boas práticas em projetos

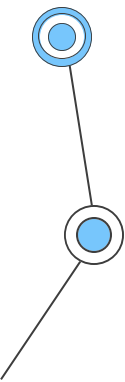
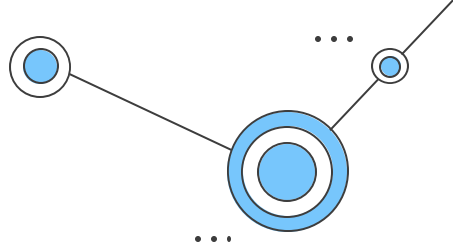
- Considere abordagens alternativas.
 - Padrões.
 - Estilos de arquitetura.
 - Tecnologias.
- Ser rastreável ao modelo de análise.
 - Consistência entre modelos os artefatos.
 - Facilita o entendimento.



Introdução a Projeto de Software

Boas práticas em projetos

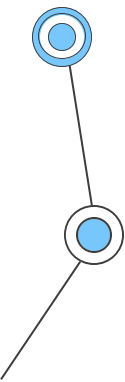
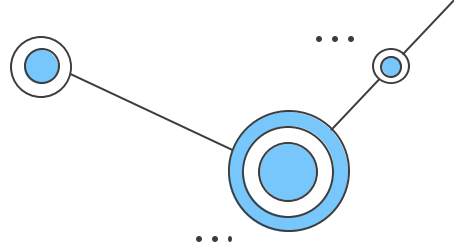
- Minimize a distância entre o software e o problema do mundo real.
 - Software inovador.
 - Diferença para o negócio.
- Considere os aspectos operacionais da aplicação.
 - Determine métricas operacionais.
 - Estabeleça níveis de dependências.



Introdução a Projeto de Software

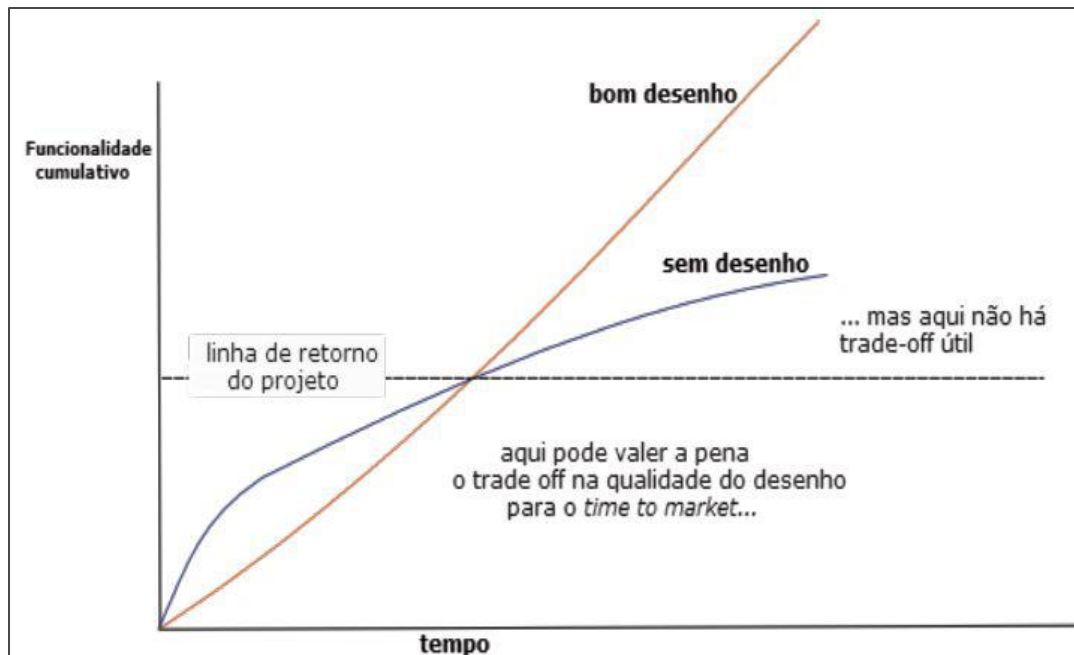
Boas práticas em projetos

- Organize-se para gestão da qualidade.
 - Estruture para acomodar mudanças.
 - Use técnicas e ferramentas automatizadas de controle de qualidade.
 - Use testes unitários, análise de dependência, análise de código estático ...
 - Defina métricas comportamentais e de desempenho para componentes e subsistemas.



Introdução a Projeto de Software

Bom projeto é investimento



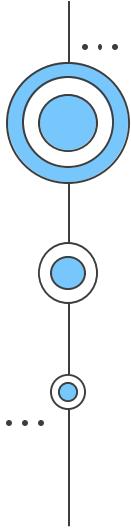
Fonte: FOWLER, M. Design Stamina Hypothesis.

Introdução a Projeto de Software

Conclusão

- Projeto é o que quase todo engenheiro de software quer fazer.
- É o lugar onde a criatividade impera.

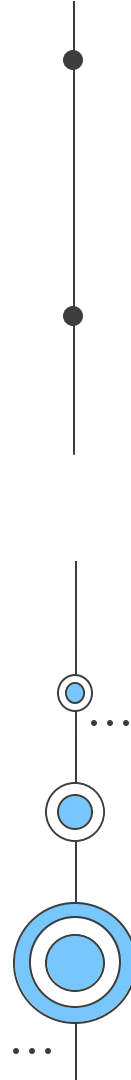


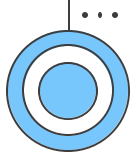


Projeto de Software

Referências básicas:

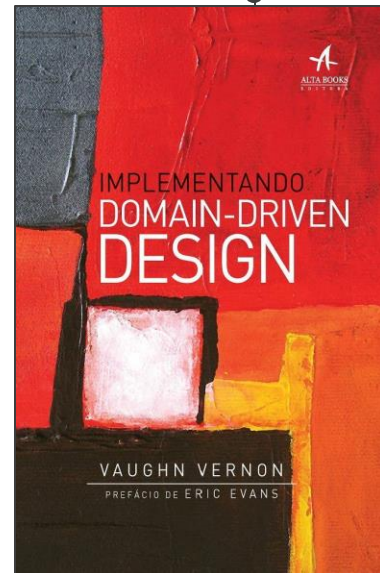
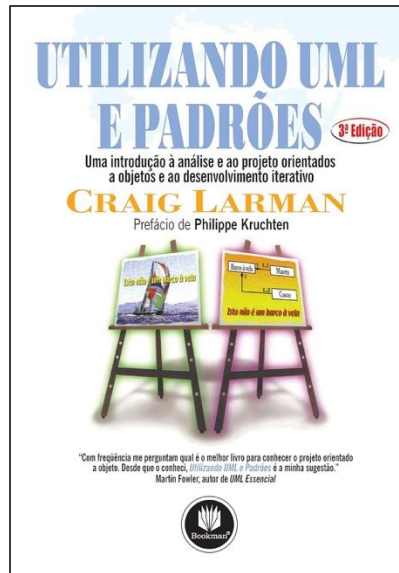
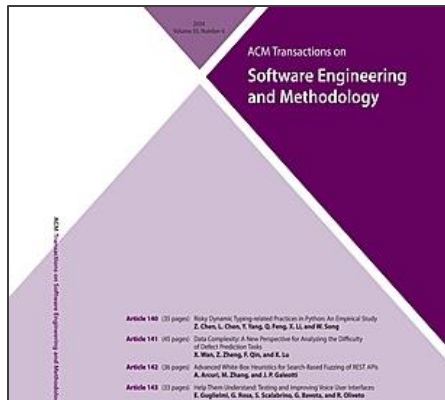
- **ACM TRANSACTIONS ON SOFTWARE ENGINEERING AND METHODOLOGY**. New York, N.Y., USA: Association for Computing Machinery, 1992-. Trimestral. ISSN 1049-331X. Disponível em: <https://dl.acm.org/toc/tosem/1992/1/2>. Acesso em: 19 jul. 2024. (Periódico On-line).
- LARMAN, Craig. **Utilizando UML e padrões**: uma introdução á análise e ao projeto orientados a objetos e desenvolvimento iterativo. 3. ed. Porto Alegre: Bookman, 2007. E-book. ISBN 9788577800476. (Livro Eletrônico).
- SILVEIRA, Paulo et al. **Introdução à arquitetura e design de software**: uma visão sobre a plataforma Java. Rio de Janeiro, RJ: Elsevier, Campus, 2012. xvi, 257 p. ISBN 9788535250299. (Disponível no Acervo).
- VERNON, Vaughn. **Implementando o Domain-Driven Design**. Rio de Janeiro, RJ: Alta Books, 2016. 628 p. ISBN 9788576089520. (Disponível no Acervo).



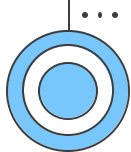


Projeto de Software

Referências básicas:



...

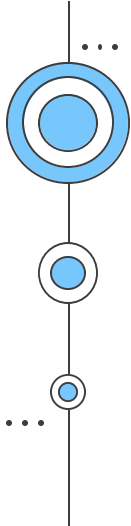


Projeto de Software

Referências complementares:

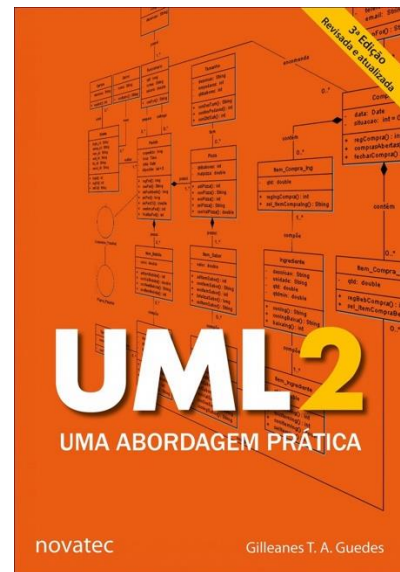
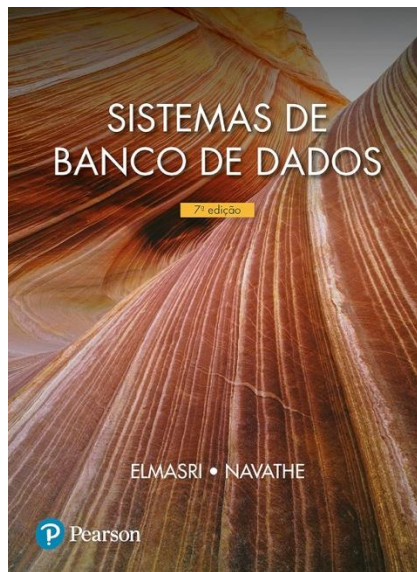
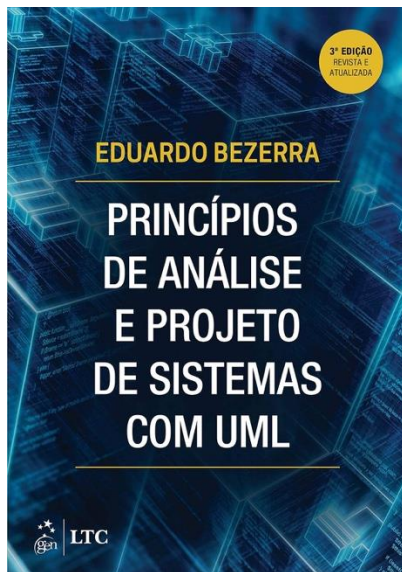
- BEZERRA, Eduardo. **Princípios de análise e projeto de sistemas com UML**. 3. ed. rev. e atual. Rio de Janeiro: Elsevier, 2015. xvii, 398 p. ISBN 9788535226263. (Disponível no Acervo).
- ELMASRI, Ramez; Navathe, Shamkant B. **Sistemas de banco de dados**, 7ª ed. Editora Pearson 1152 ISBN 9788543025001. (Livro Eletrônico).
- GUEDES, Gilleanes T. A. **UML 2**: uma abordagem prática. 2. ed. São Paulo: Novatec, c2011. 484 p. ISBN 9788575222812. (Disponível no Acervo).
- **IEEE TRANSACTIONS ON SOFTWARE ENGINEERING**. New York: IEEE Computer Society, 1975-. Mensal,. ISSN 0098-5589. Disponível em: <https://ieeexplore.ieee.org/xpl/RecentIssue.jsp?punumber=32>. Acesso em: 19 jul. 2024. (Periódico On-line).
- SOMMERVILLE, Ian. **Engenharia de software**. 10. ed. São Paulo: Pearson Education do Brasil, c2019. xii, 756 p. ISBN 9788543024974. (Disponível no Acervo).
- WAZLAWICK, Raul Sidnei. **Análise e design orientados a objetos para sistemas de informação**: modelagem com UML, OCL e IFML. 3. ed. Rio de Janeiro, RJ: Elsevier, Campus, c2015. 462 p. ISBN 9788535279849. (Disponível no Acervo).



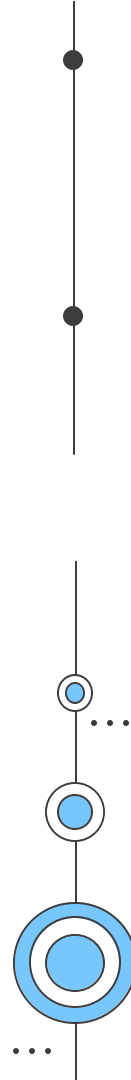


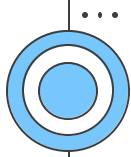
Projeto de Software

Referências complementares:



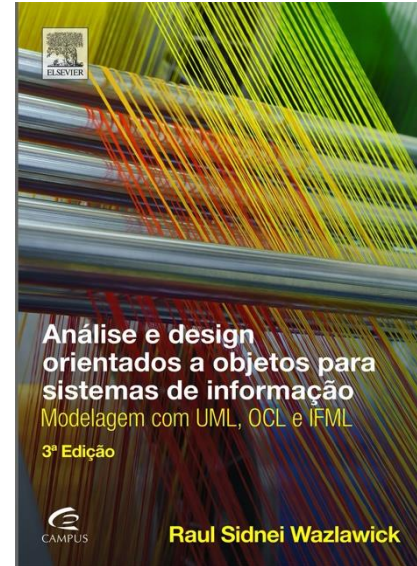
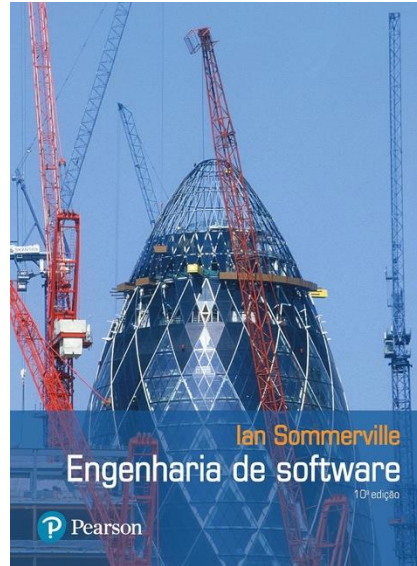
...





Projeto de Software

Referências complementares:



Obrigado!

Dúvidas?

joaopauloaramuni@gmail.com



[GitHub](#)



[LinkedIn](#)



[Lattes](#)

...