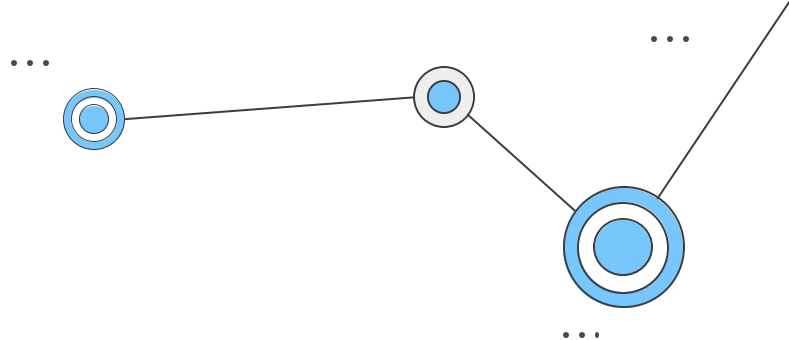
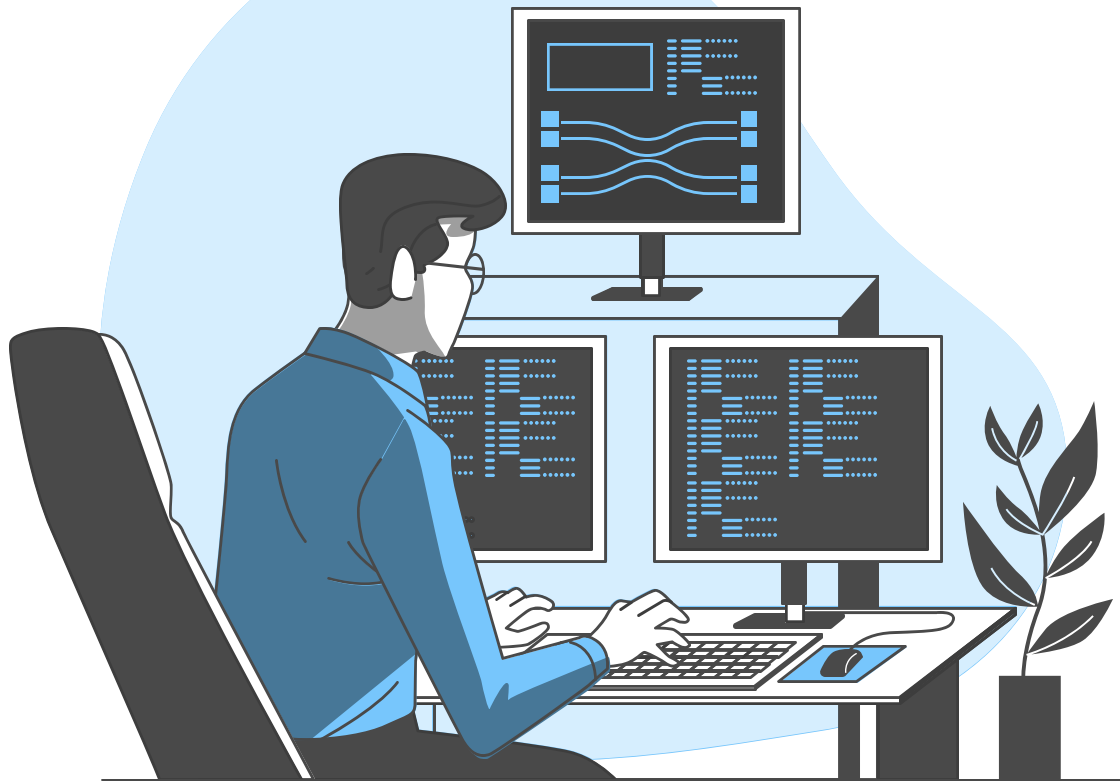




PUC Minas



# Projeto de Software

Prof. Dr. João Paulo Aramuni



# Unidade 1

## Introdução

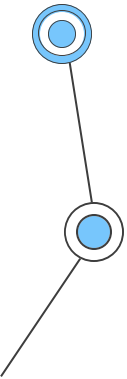
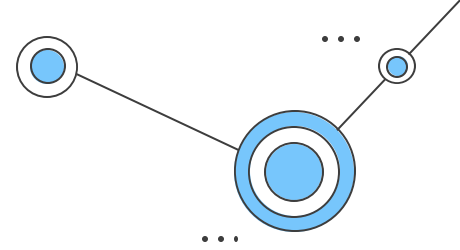
PDS - Manhã /Noite



# Introdução a Projeto de Software

Conceitos básicos aplicado a projetos: **Independência funcional**

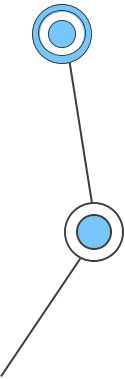
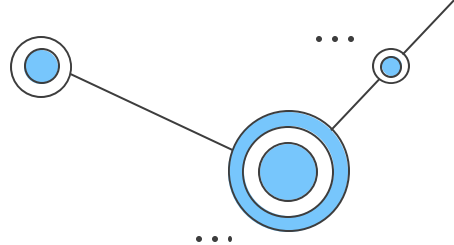
- A independência funcional é alcançada através do desenvolvimento de módulos simples e uma "aversão" à interação excessiva com outros módulos.
- **Coesão** é uma indicação da força relativa funcional de um módulo (realiza uma única tarefa).
- **Acoplamento** é uma indicação da relação entre os módulos interdependência (buscar o menor grau de acoplamento possível).



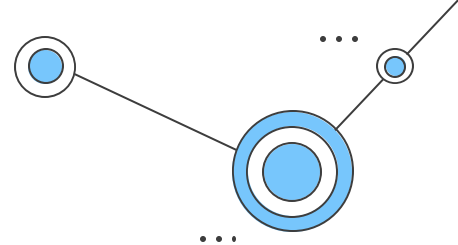
# Introdução a Projeto de Software

Conceitos básicos aplicado a projetos: **Refinamento**

- Uma elaboração maior no nível de detalhes.
- Um programa é desenvolvido através de sucessivos refinamentos.
  - Chegamos a uma descrição em linguagem de programação partindo de uma em linguagem natural e refinando-a em sucessivas iterações.
- Abstração e refinamento são conceitos complementares.



# Introdução a Projeto de Software



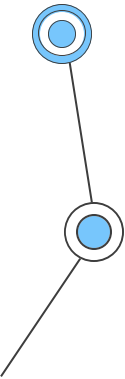
## Refatoração

Fowler define refatoração da seguinte maneira:

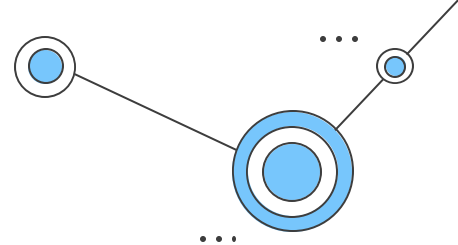
- "Refatoração é o processo de mudança de um sistema de software de tal maneira que não altere o comportamento externo do código [design] e ainda melhora a sua estrutura interna."

Quando o software é reformulado, o projeto existente é examinado para verificar:

- Redundância
- elementos de design não utilizados
- algoritmos ineficientes ou desnecessários
- estruturas de dados inadequadas ou mal construída
- ou qualquer outra falha de design que pode ser corrigido para produzir um design melhor..



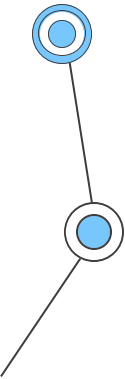
# Introdução a Projeto de Software



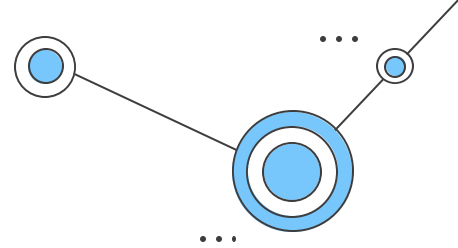
Conceitos básicos aplicado a projetos: **Independência Funcional**

Modularidade + abstração + ocultação = Independência Funcional

- “Finalidade única” e menos interação
  - Interfaces simplificadas
  - Manutenção mais fácil
  - Propagação de erros minimizada
  - Reutilização
- 
- Dois critérios (qualitativos) para avaliação:
  - ✓ COESÃO: robustez funcional de um módulo (módulo realiza uma única tarefa).
  - ✓ ACOPLAMENTO: indicação da interdependência entre módulos.



# Introdução a Projeto de Software



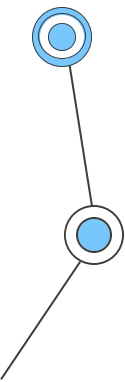
Conceitos básicos aplicado a projetos: **Refinamento e Refatoração**

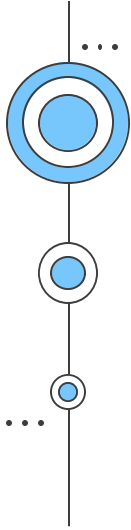
## Refinamento

- Processo de elaboração (alto nível -> mais detalhes).
- Refinamentos sucessivos.
- (Abstração + refinamentos): conceitos complementares.

## Refatoração

- Reorganizar para simplificar o projeto sem alterar as funções e os comportamentos.
- O que pode ser refatorado?
  - ✓ Redundância, elementos não utilizados, algoritmos ineficientes, etc.

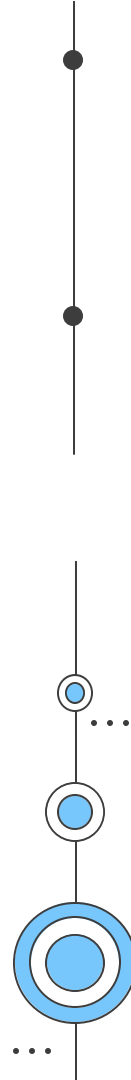




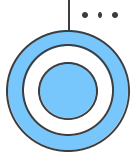
# Projeto de Software

## 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).

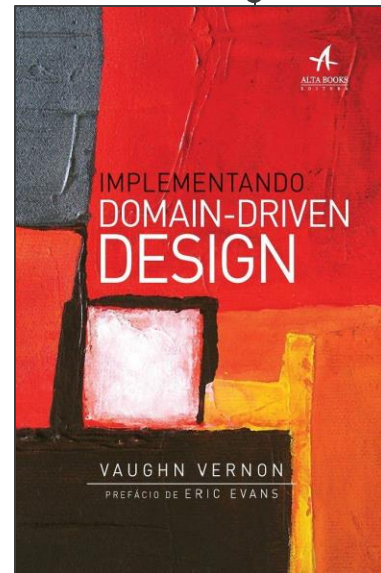
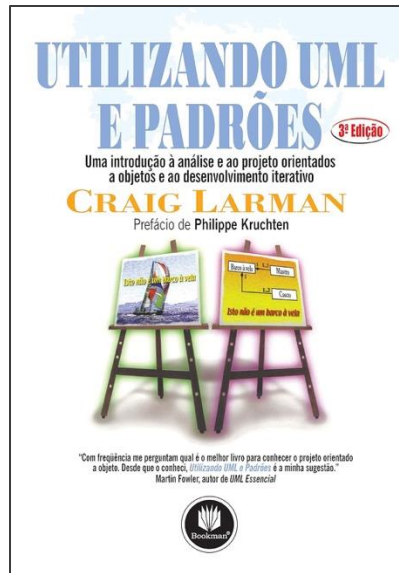
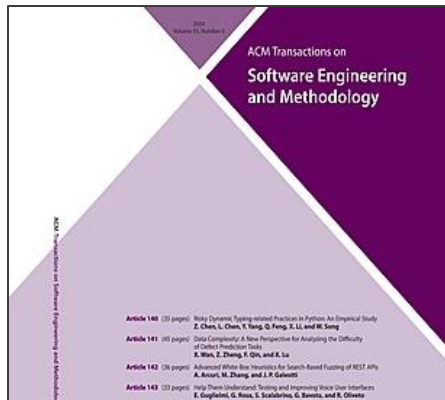




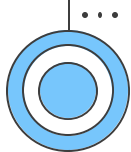


# Projeto de Software

## Referências básicas:



...

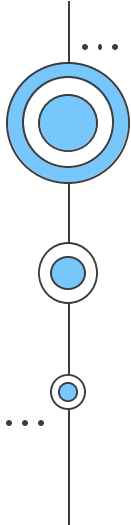


# Projeto de Software

## Referências complementares:

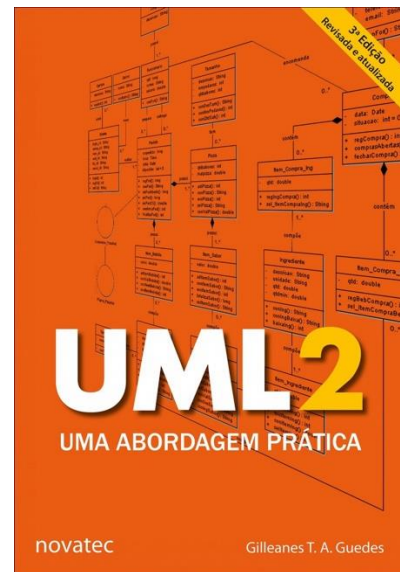
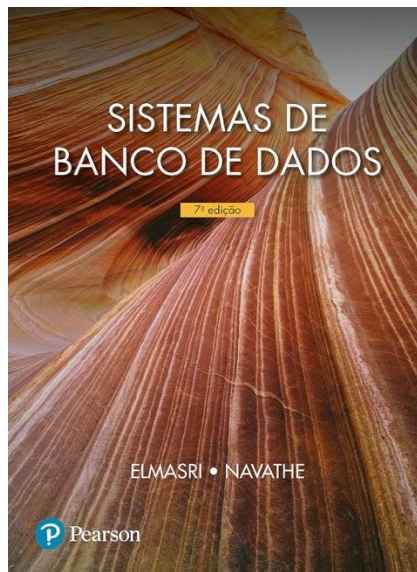
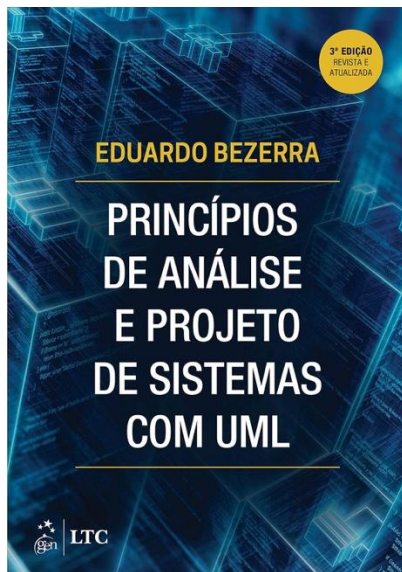
- BEZERRA, Eduardo. **Princípios de análise e projeto de sistemas com UML**. 3. ed. rev. e atual. Rio de Janeiro: Elsevier, 2015. xvii, 398 p. ISBN 9788535226263. (Disponível no Acervo).
- ELMASRI, Ramez; Navathe, Shamkant B. **Sistemas de banco de dados**, 7ª ed. Editora Pearson 1152 ISBN 9788543025001. (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).



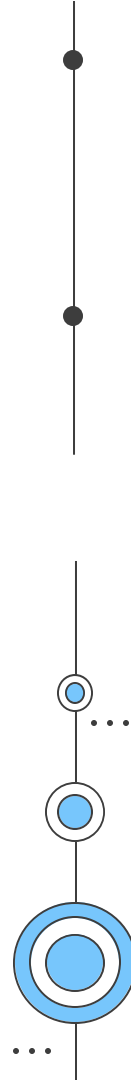


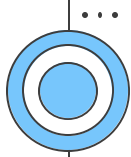
# Projeto de Software

## Referências complementares:



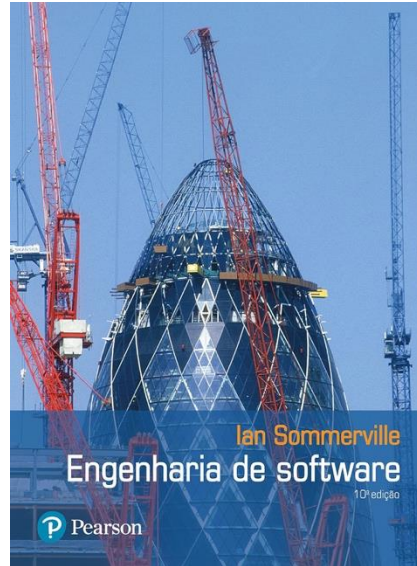
...





# Projeto de Software

## Referências complementares:



# Obrigado!

Dúvidas?

joaopauloaramuni@gmail.com



[GitHub](#)



[LinkedIn](#)



[Lattes](#)

...