

Capítulo 7: Engenharia de Software

**Ciência da computação: Uma visão abrangente
11a Edição**

**by
J. Glenn Brookshear**

Capítulo 7: Engenharia de software

- 7.1 A disciplina de Engenharia de Software
- 7.2 O ciclo de vida de Software
- 7.3 Metodologias de Engenharia de Software
- 7.4 Modularidade
- 7.5 Ferramentas do comércio
- 7.6 Teste
- 7.7 Documentação
- 7.8 Responsabilidade e Propriedade do software

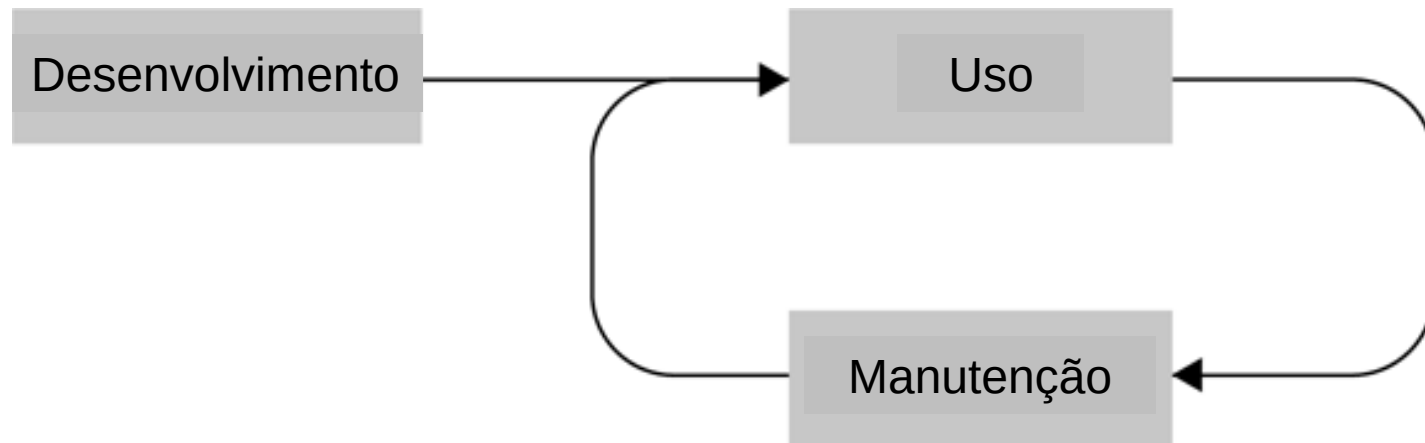
A disciplina de Engenharia de Software

- Distinta de outras áreas de engenharia
 - Componentes pré-fabricados
 - Métricas
- Praticantes contra teóricos
- Organizações profissionais: ACM, IEEE, etc.
 - Códigos de ética profissional
 - Normas

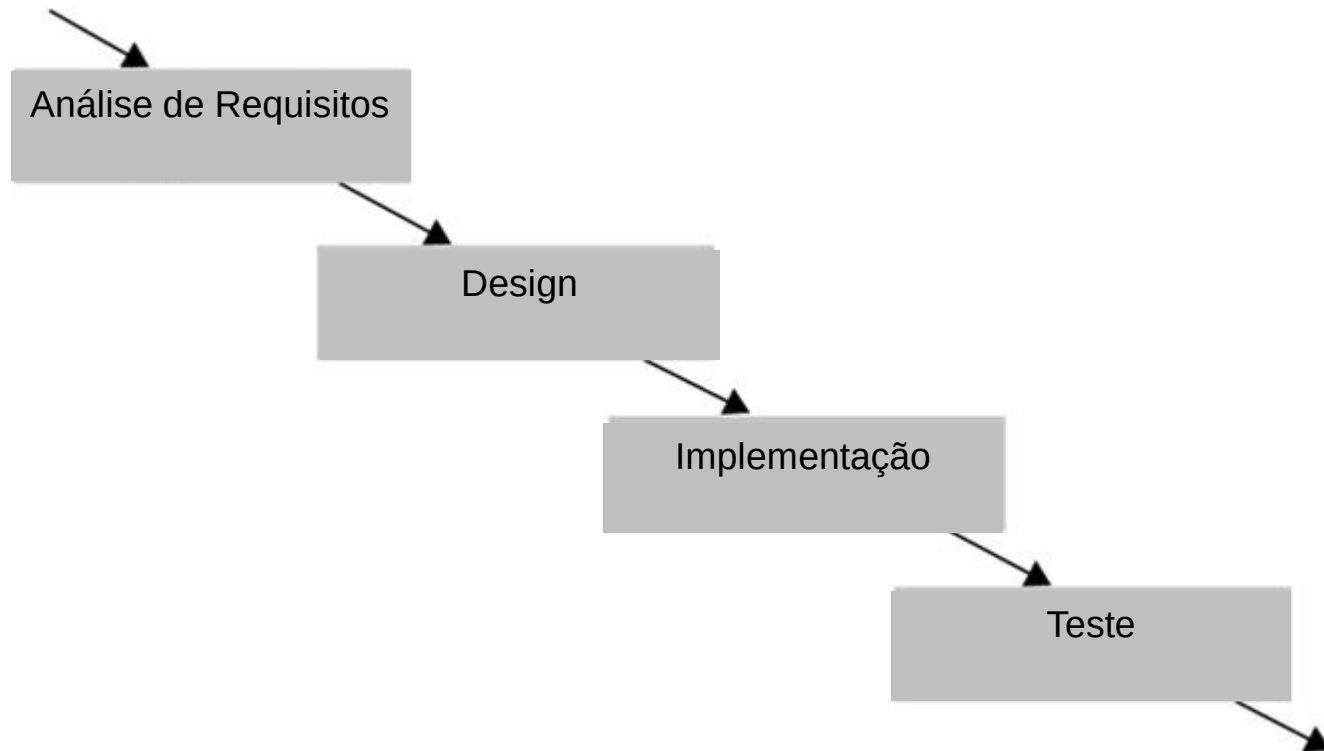
Computer Aided Software Engineering (CASE) tools

- Planejamento de projeto
- Gerenciamento de projetos
- Documentação
- Prototipagem e simulação
- Design de interface
- Programação

O ciclo de vida de software



A fase de desenvolvimento do ciclo de vida de software



Etapa de Análise de Requisitos

- Requisitos
 - Orientado à Aplicação
- Especificações
 - Orientado tecnicamente
- Documento de requisitos de software

Fase de Concepção (Design) - Projeto

- Metodologias e ferramentas (discutidas mais tarde)
- Interface humana (Psicologia e ergonomia)

Fase de implementação

- Criar o sistema a partir de um projeto
 - Escrever programas
 - Criar arquivos de dados
 - Desenvolver bases de dados
- Papel de "analista de software" versus "programador"

Fase de testes

- Testes de validação
 - Confirmar que o sistema atende as especificações
- Testes de defeitos
 - Encontrar bugs

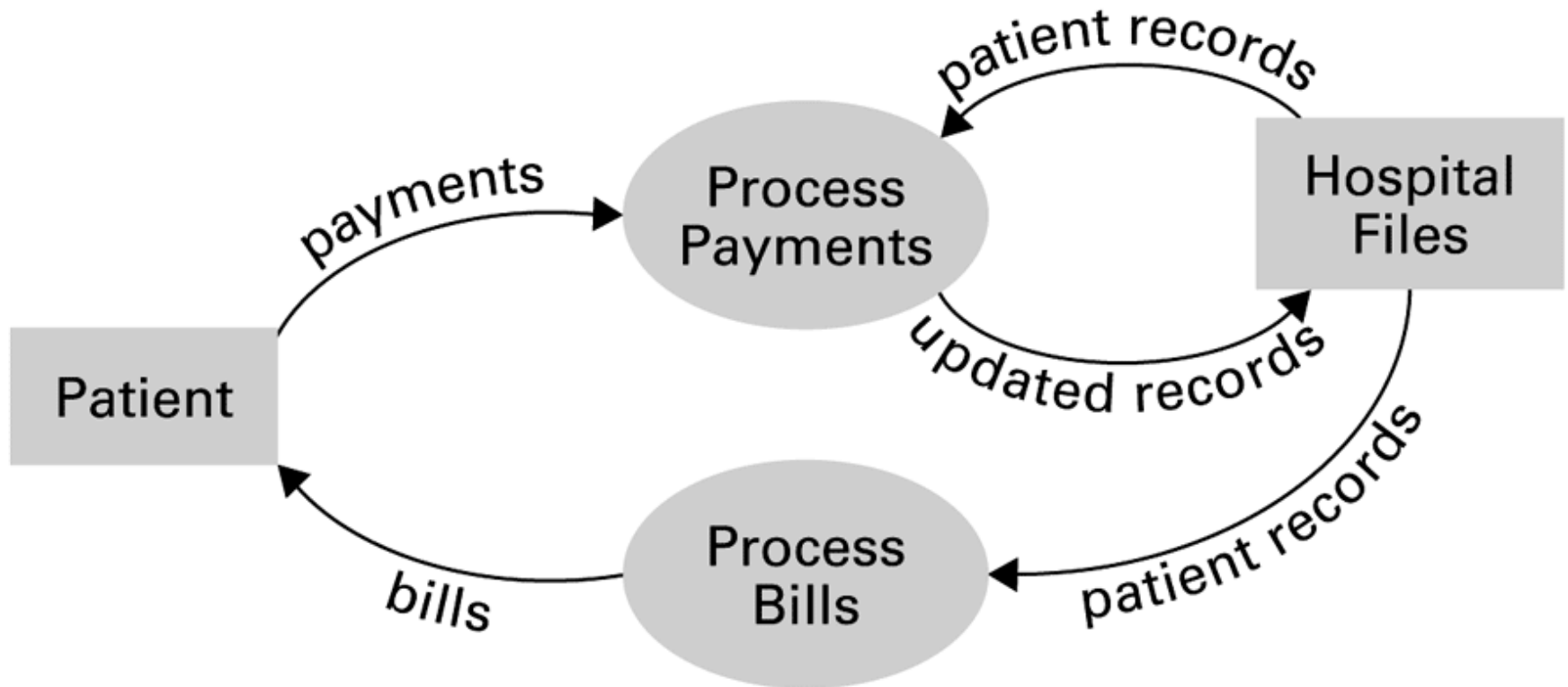
Metodologias de Engenharia de Software

- Modelo em cascata
- Modelo Incremental
 - Prototipagem (Evolucionária vs. Descartável)
- Desenvolvimento em código aberto
- *Extreme Programming*

Ferramentas

- Diagrama de fluxo de dados
- Diagrama entidade-relacionamento
 - Relação um para um
 - Relação um-para-muitos
 - Relação muitos-para-muitos
- Dicionário de dados
- UML – Unified Modeling Language
 - Diagrama de Casos de Uso
 - Diagrama de Classes

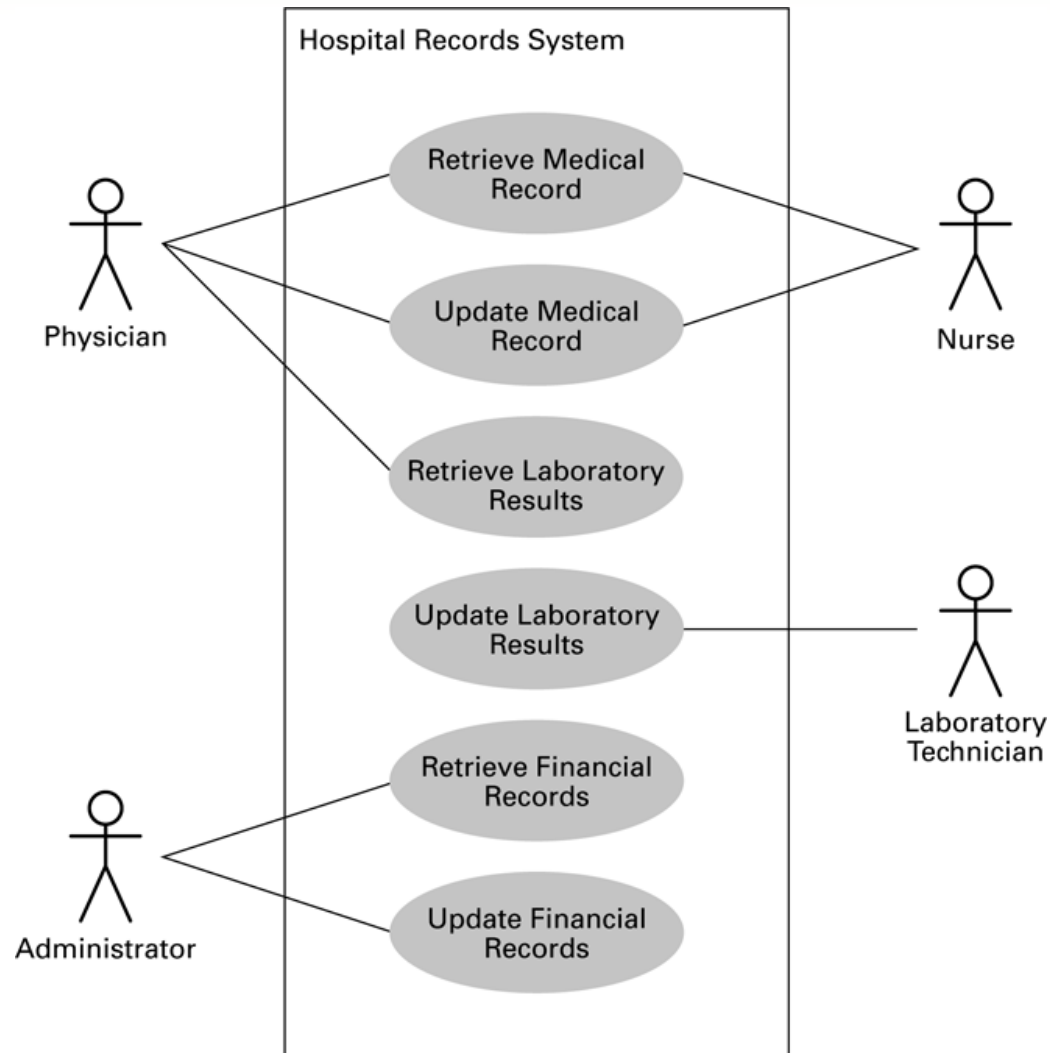
Um diagrama de fluxo de dados simples



UML - Unified Modeling Language

- Diagrama de caso de uso
 - Casos de uso
 - Atores
- Diagrama de classes

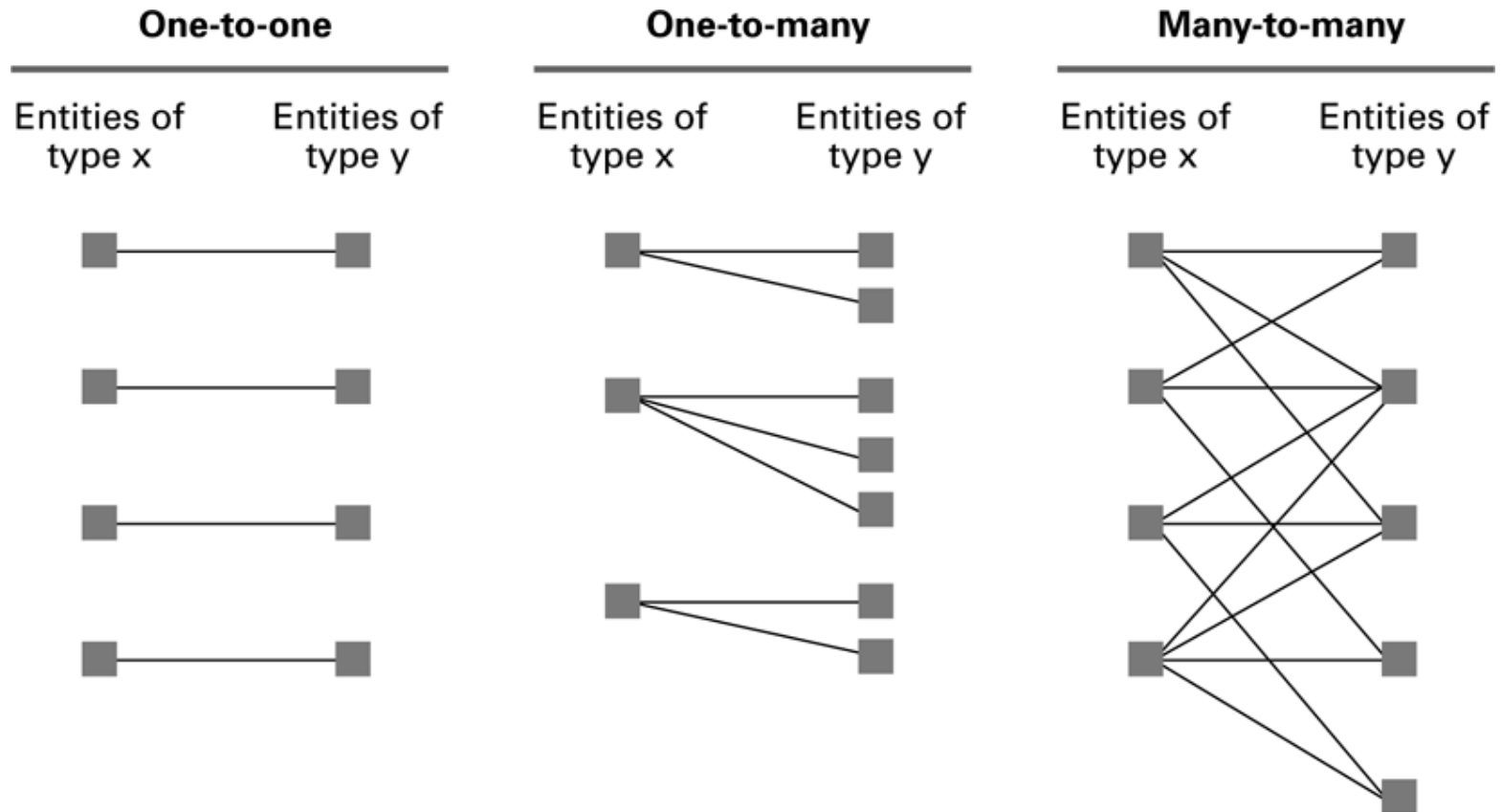
Um diagrama de caso de uso simples



Um diagrama de classe simples



Relacionamentos um-para-um, um-para-muitos e muitos-para-muitos entre entidades dos tipos X e Y



Um diagrama de classe retratando generalizações

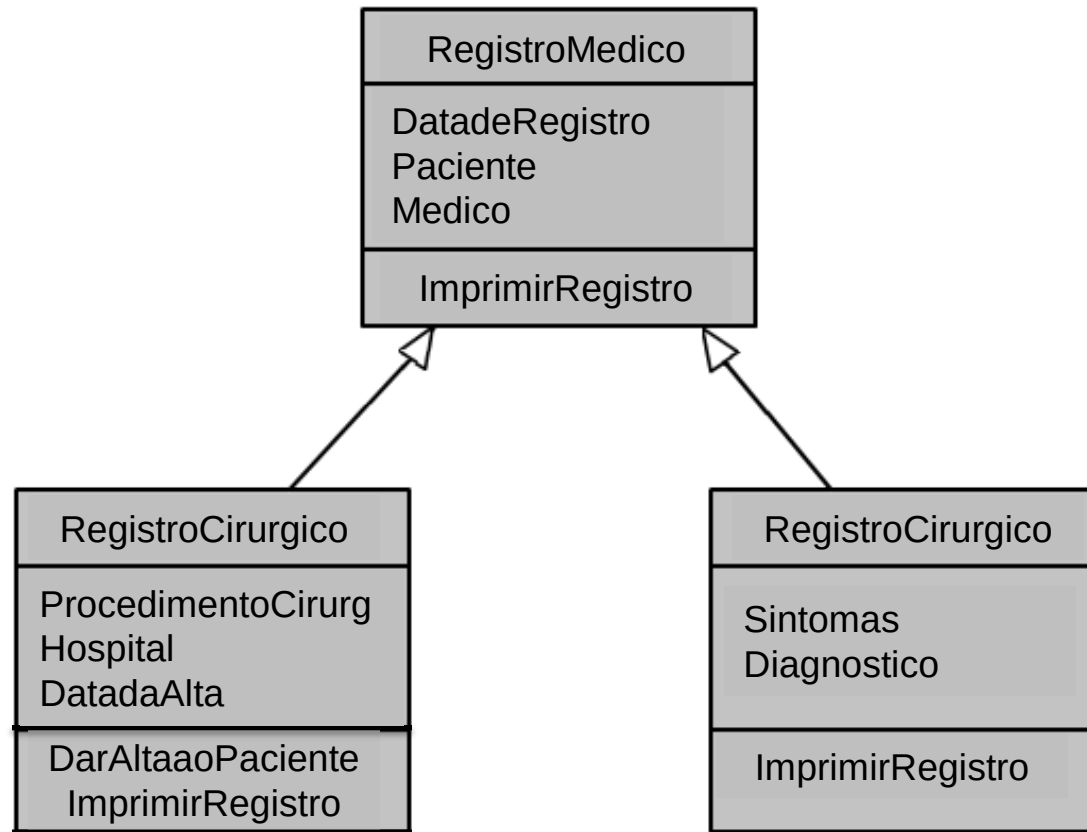
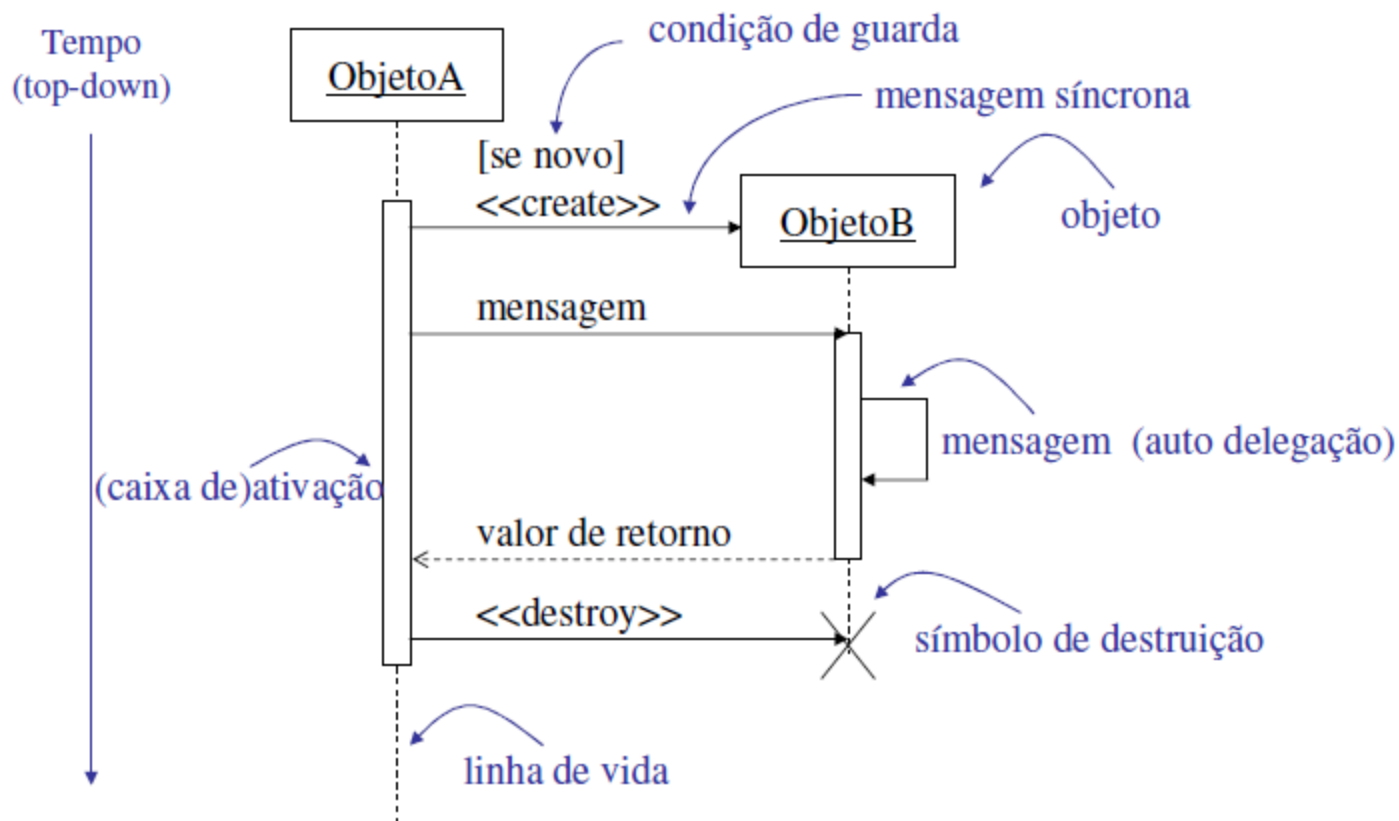
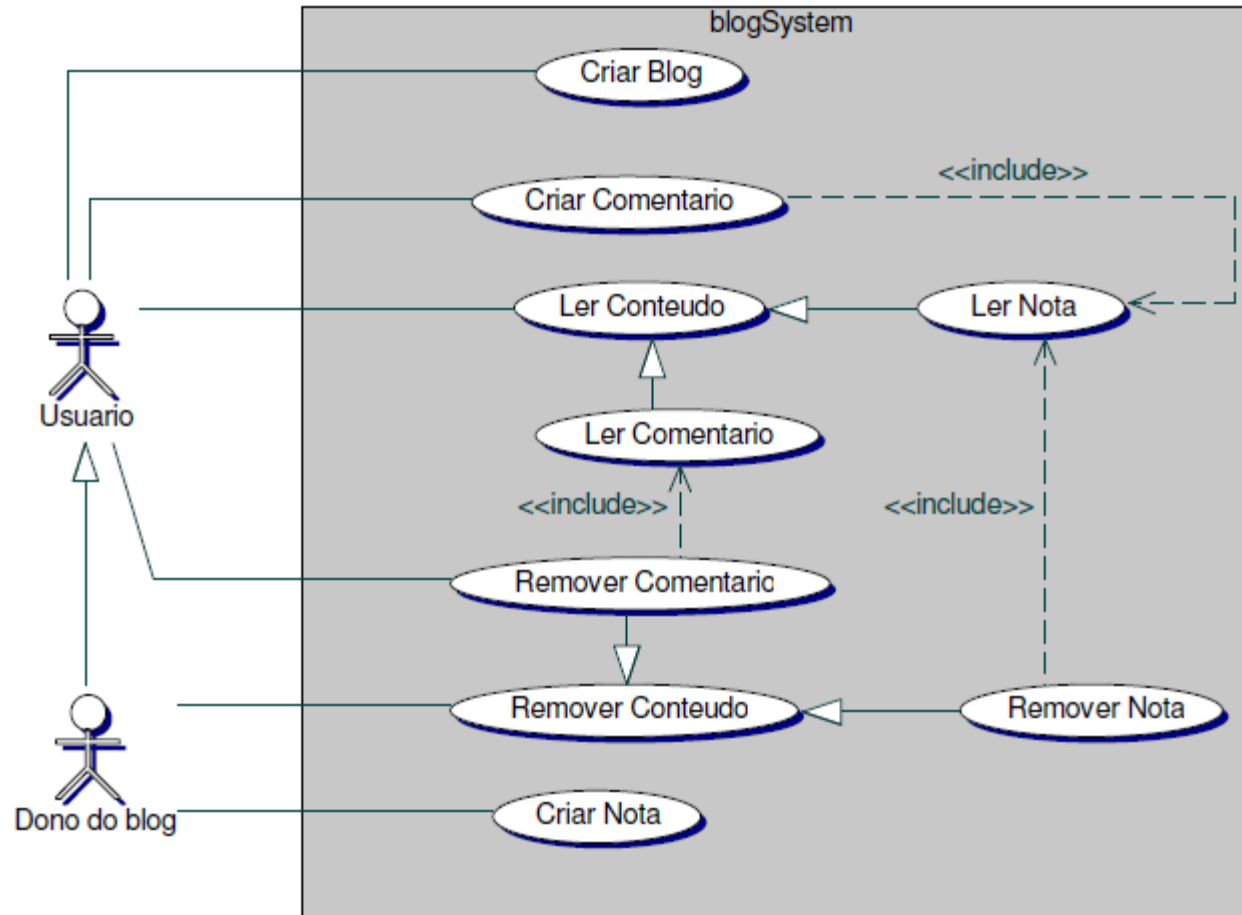


Diagrama de seqüência



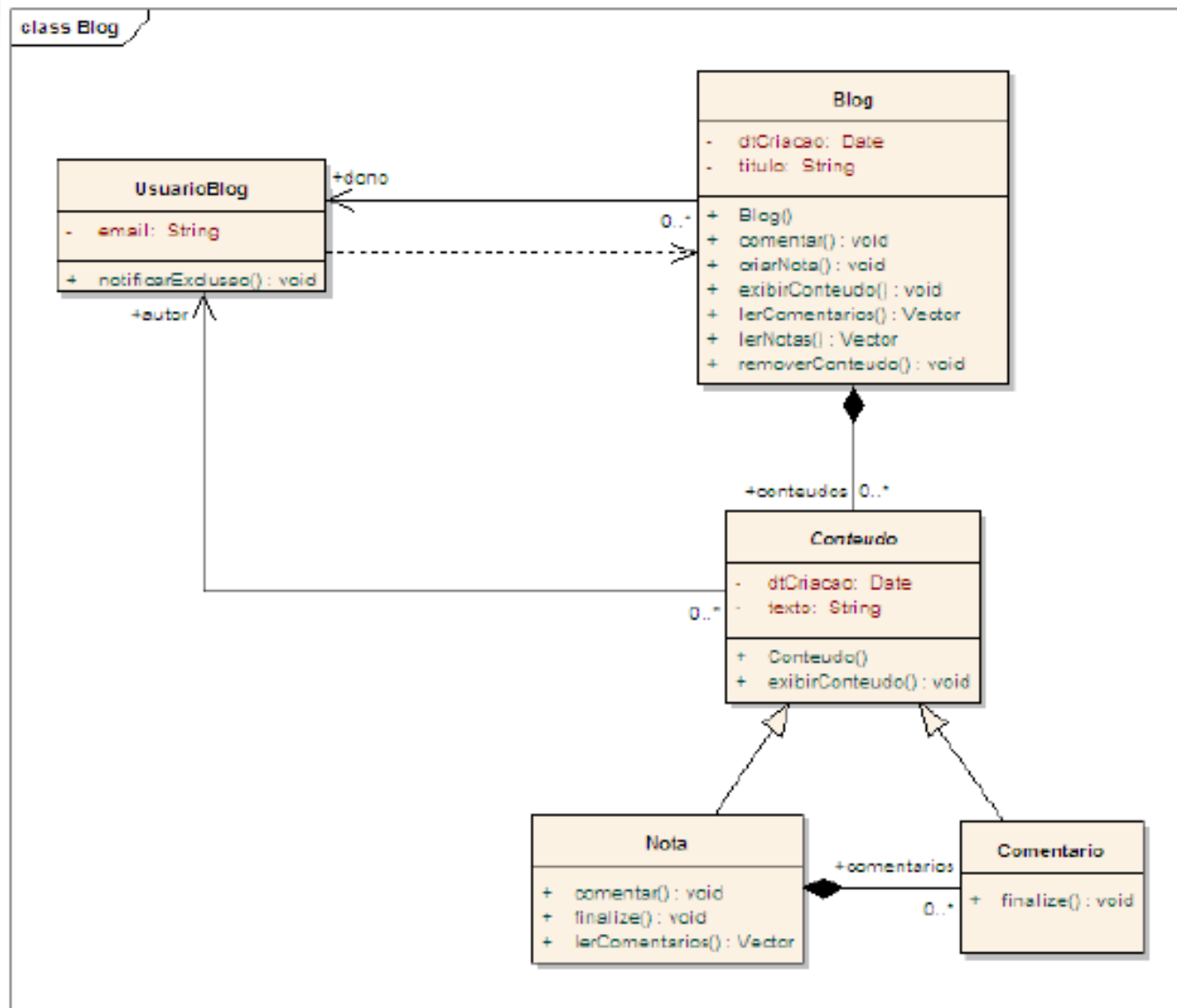
Fonte: LES (PUC-Rio)

Exemplo: Blog (Fonte: LES – PUC-Rio)

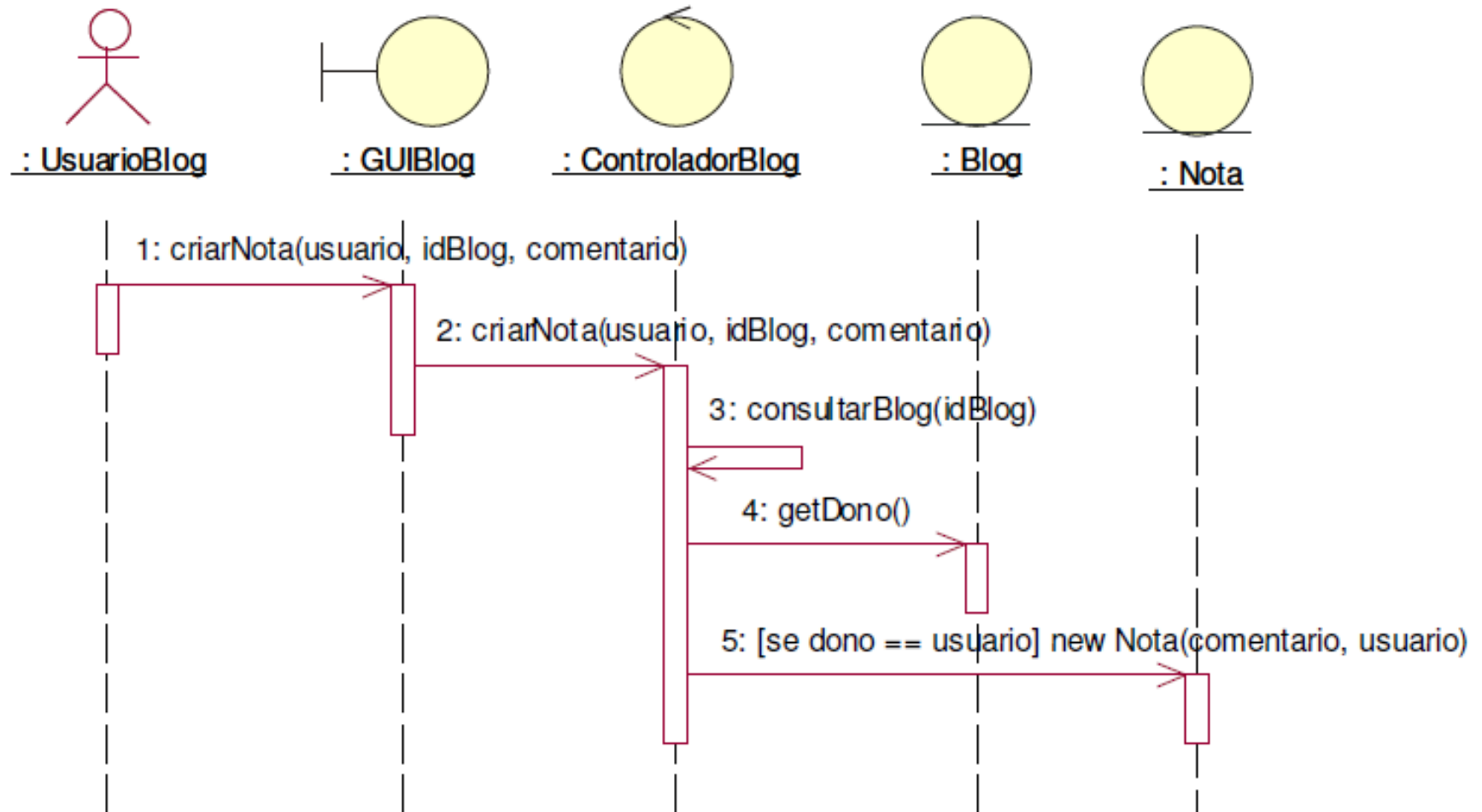


Caso de Uso

Blog: Diagrama de Classes



Blog: Diagrama de Sequencia de “criar nota”



Design Patterns

- "Modelos" bem projetados para resolver problemas recorrentes
- Exemplos:
 - Adapter pattern: Usado para adaptar a interface de um modulo às necessidades atuais
 - Decorator pattern: Usado para controlar a complexidade envolvida quando muitas combinações diferentes das mesmas atividades são necessárias
- Inspirados no trabalho de Christopher Alexander em arquitetura

Estratégias de teste de software

- Teste Glass-box (caixa branca)
 - Princípio de Pareto
 - Erros de software tendem a ser concentrados, e não dispersos. Assim, resultados frequentemente podem ser melhorados aplicando-se os esforços em uma área específica
 - Teste de caminho básico
 - Cada instrução deve ser executada ao menos 1 vez
- Teste Black-box (caixa preta)
 - Análise dos valores limite
 - Teste de redundância
 - Beta testing

Documentação

- Documentação de usuário
 - Livro impresso para todos os clientes
 - Módulos de ajuda on-line
- Documentação do sistema
 - Código-fonte
 - Documentos de projeto
- Documentação técnica
 - Para instalar, personalizar, atualizar, etc.

Propriedade do software

- Copyright
 - Permitir que um produto seja lançado, mantendo a posse da propriedade intelectual
 - Garantia em todas as partes do produto:
 - Especificações
 - Código-fonte
 - Produto final

Propriedade do software (continuação)

- Licença de software
 - Um acordo legal que concede ao usuário determinadas permissões sem transferência de propriedade
- Patentes
 - Deve demonstrar que é novo, útil e não óbvio para os outros com origens semelhantes
 - Process is expensive and time-consuming