

# **PÓS-GRADUAÇÃO EM ENGENHARIA DE SOFTWARE**

**Planejamento e Realização de Teste de Software**


## **Aula 07 Teste de Arquitetura de Sistemas**

Prof. Rogério Messias

# Pauta da aula

- › Visão Geral de Arquitetura de Software
- › IEEE 1471/2000
- › Cenário de Atributos de Qualidade
- › Cenários Genéricos
- › Cenários de 6 partes
- › Cenários Específicos

# Atividades de testes associadas às fases de desenvolvimento

Fase	Atividade
Engenharia de Requisitos	<ul style="list-style-type: none"><li>• Determinar as estratégias de testes</li><li>• Especificação de requisitos de testes</li><li>• Geração de dados de testes funcionais</li></ul>
Projeto	 <ul style="list-style-type: none"><li>• Checar a consistência entre a especificação de requisitos e o projeto (design)</li><li>• Avaliar a arquitetura de software</li><li>• Testar o projeto de arquitetura</li><li>• Geração de dados de teste funcional e estrutural</li></ul>
Implementação	<ul style="list-style-type: none"><li>• Verificar a consistência entre o projeto (design) e a implementação</li><li>• Testar o programa</li></ul>
Manutenção	<ul style="list-style-type: none"><li>• Repetir os testes acima de acordo com o nível de retrabalho.</li></ul>

Fonte: VLIET, 2008

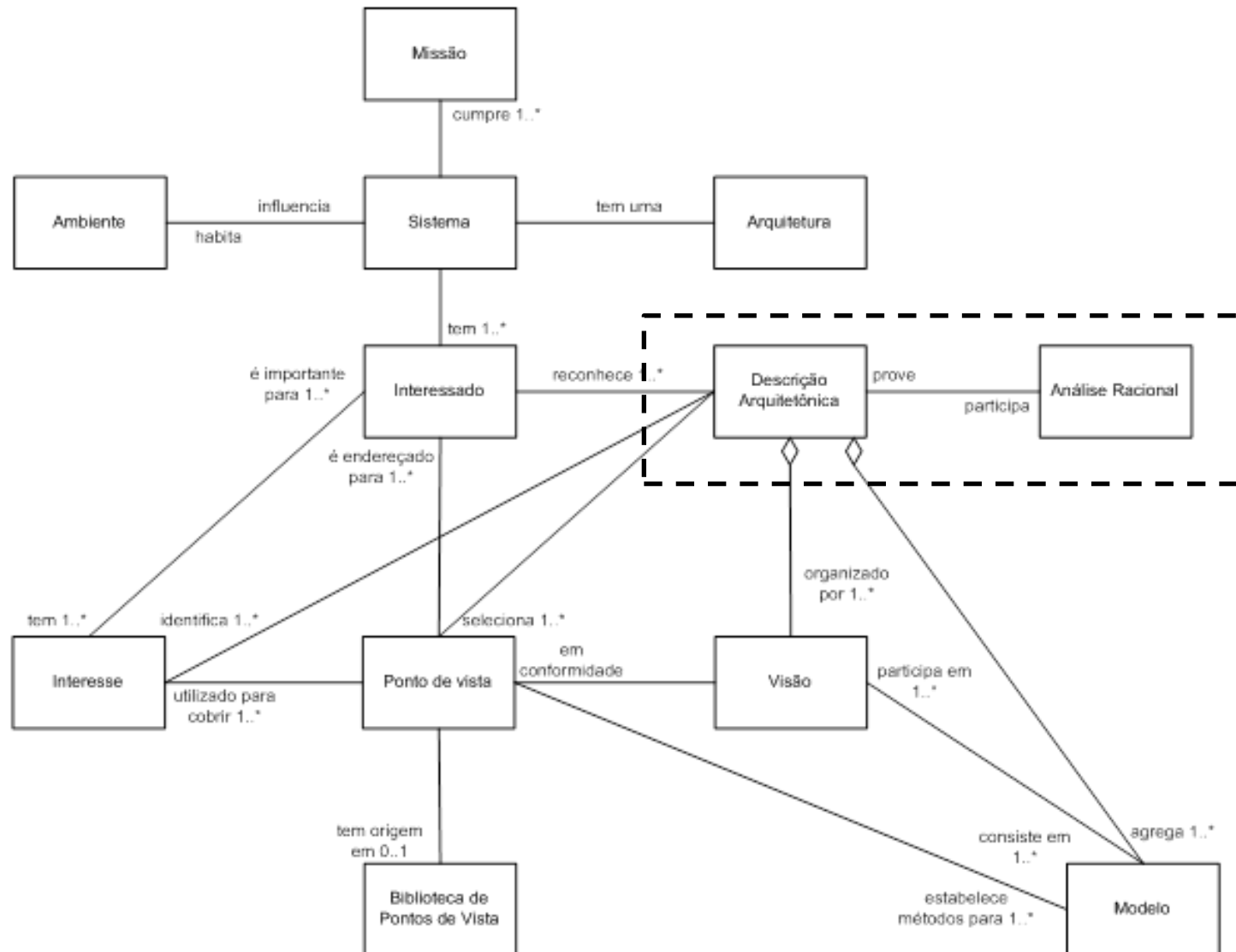
# Afinal de contas o que é arquitetura?

- › Segmento guiado por decisões técnicas e comerciais.
- › Visa assegurar a satisfação dos:
  - › Requisitos de Funcionais (Comportamento)
  - › Requisitos de Não Funcionais (Atributos de Qualidade)
  - › Metas de Negócio
- › É direcionada tanto por aspectos operacionais, considerando requisitos funcionais e não funcionais, como por aspectos organizacionais, considerando o contexto da organização que desenvolve o sistema.

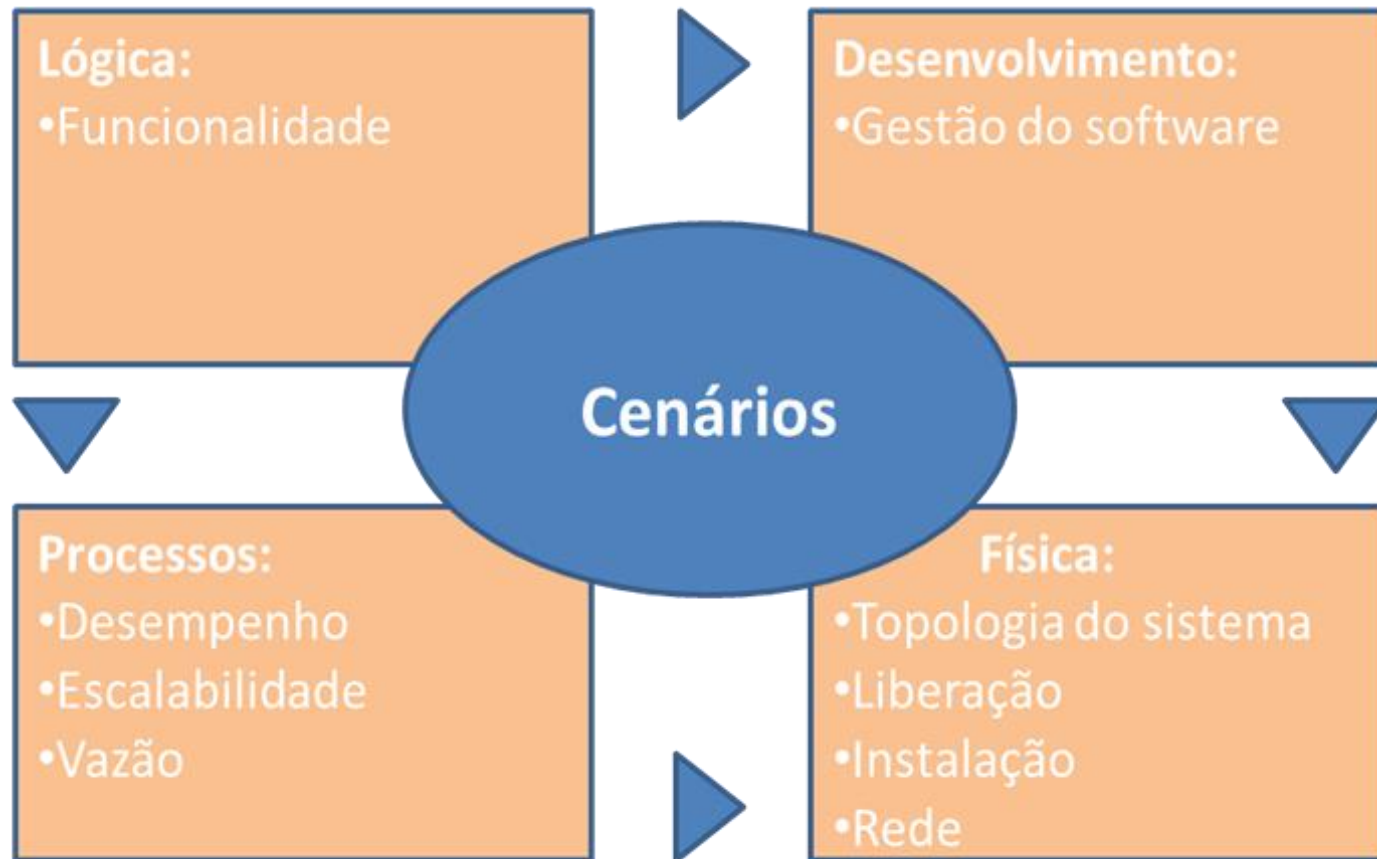
(KRUCHTEN, 2003)
- › Estrutura composta por elementos de software, propriedades visíveis e externas e o relacionamento entre estes elementos.

(BASS, 2003)

# IEEE 1471/2000 - Prática recomendada para descrição de arquitetura de sistemas

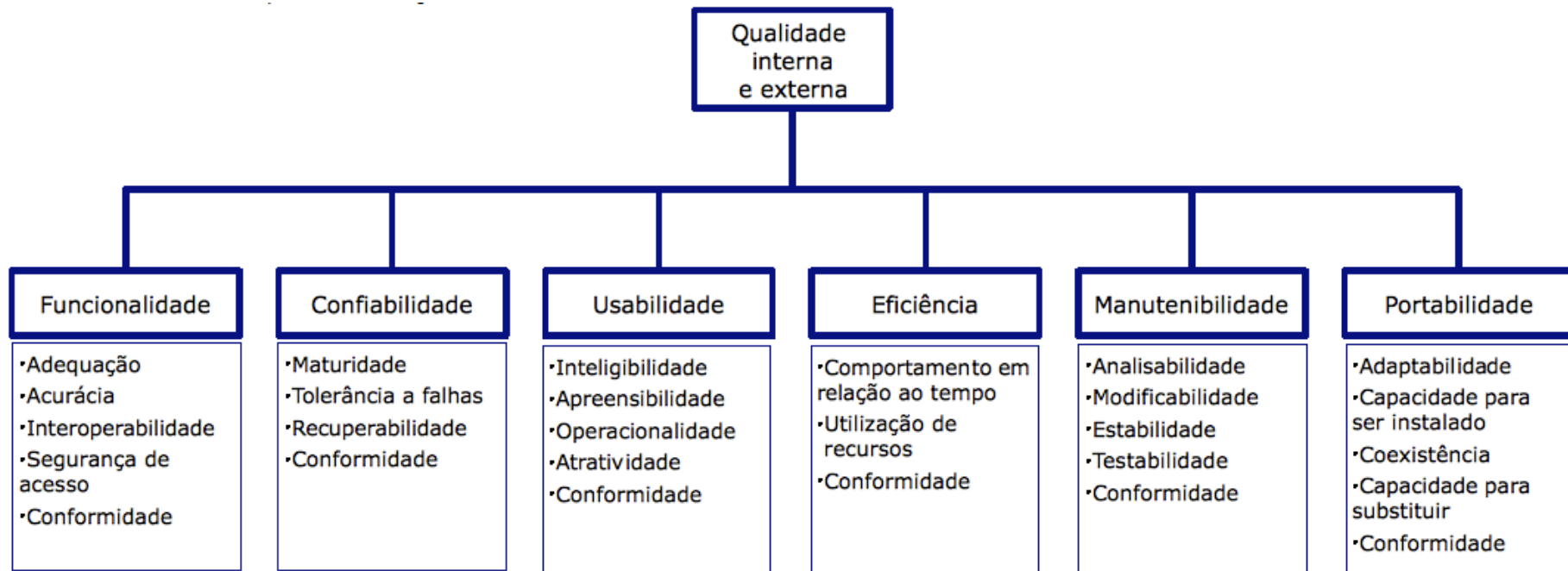


# Avaliação da arquitetura de software



(KRUCHTEN, 2003)

# ISSO/IEC 9126 - Atributos de Qualidade

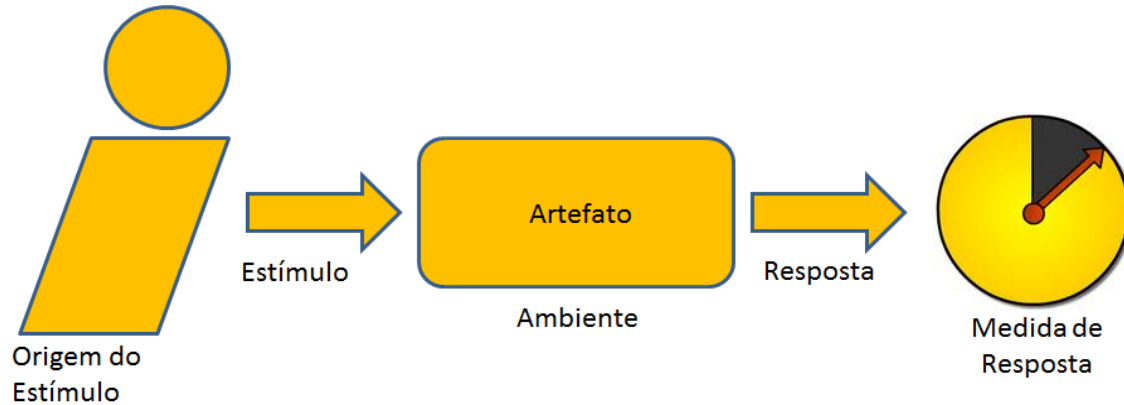


# Cenários de Qualidade

- › Um cenário deve descrever como o sistema satisfaz um determinado atributo de qualidade.
- › Deve usar um modelo ou estrutura de linguagem que possibilite a sua avaliação:
  - › Ao determinar um nível de excelência a ser atingido, torna-se possível avaliar o comportamento do sistema para verificar se ele atende ou não a especificação determinada.
- › Exemplo:
  - › Desempenho do sistema:
    - › Em condições normais, os usuários realizam 1000 requisições de arquivo por minuto e essas requisições devem ser processadas em um tempo máximo de 1 minuto.



# Modelo de Cenário 6 Partes



- › **Origem do Estímulo:** entidade que gera o estímulo;
- › **Estímulo:** condição que precisa ser considerada quando ocorre no sistema;
- › **Ambiente:** condição em que o sistema se encontra quando ocorre o estímulo;
- › **Artefato:** artefato que é estimulado (pode ser todo o sistema ou parte dele);
- › **Resposta:** atividades empreendidas após o surgimento do estímulo;
- › **Medida de Resposta:** forma de **medir a resposta** ao estímulo, de modo que a torne verificável.

# Cenários Genéricos de Qualidade

	Origem	Estímulo	Artefato	Ambiente	Resposta	Medida
<b>Disponibilidade</b>	Interna ou Externa	Falha	Recurso altamente requerido	Estado do sistema	Possíveis reações às falhas	Tempo associado à identificação da falha e reparo do sistema
<b>Modificabilidade</b>	Interessado	Mudança de funcionalidade	Sistema, plataforma ou ambiente	Quando a alteração pode ser feita	Ponto da arquitetura que deve ser alterado	Tempo e custo da mudança
<b>Desempenho</b>	Interna ou Externa	Eventos esporádicos, periódicos ou estocásticos.	Sistema	Normal ou sobrecarregado	Processamento do estímulo	Medidas de tempo associadas à duração do processamento
<b>Segurança</b>	Ser humano ou outro sistema	Tentativa de quebra de segurança	Serviços ou dados	Conectado ou desconectado a uma rede	Identificação e reações ao ataque	Tempo associado à identificação do ataque e ações corretivas
<b>Testabilidade</b>	Testadores	Marcos no ciclo de desenvolvimento do projeto	Parte do código ou todo o sistema	Atividade no ciclo de desenvolvimento do projeto	Acesso a estados e valores processados	Métricas obtidas a partir da verificação dos resultados de testes
<b>Usabilidade</b>	Usuário Final	Avaliação do uso e da eficiência do sistema	Sistema	Sistema em tempo de execução ou configuração	Encontro ou antecipação às necessidades do usuário	Medidas associadas à satisfação do usuário

# Disponibilidade

- › Preocupação com as possíveis falhas do sistema e suas consequências.
  - › Quanto tempo o sistema demora para se recuperar em caso de erro?
  - › Quanto tempo a sua empresa pode deixar de disponibilizar seus serviços?
- › Estímulos:
  - › Omissão: Componente falha ao responder uma requisição;
  - › Crash: Componente repetidamente sofre falhas de omissão;
  - › Cronometragem: Componente responde, mas atrasado;
  - › Resposta: Componente responde, mas com um valor incorreto.
- › Respostas:
  - › Existem muitas formas do sistema responder ao erro: Geração de log por erro, notificação de usuários do sistema, alterar o comportamento do sistema para que trabalhe em modo degradado, desligar sistemas externos.
- › Medida de Resposta:
  - › Pode ser especificada uma porcentagem de disponibilidade ou um tempo para o reparo, número de vezes que o sistema pode se tornar indisponível, a duração que o sistema deve se manter disponível.

# Disponibilidade - Exemplo

- › Desejo:
  - › O dono do sistema espera que ele esteja disponível em 99% do tempo.
- › Cenário de Disponibilidade:
  - › Caso o sistema venha a apresentar erros de cálculos internos, ele deverá tornar-se indisponível e colocado novamente em produção em menos de 2 horas.
- › Sua empresa deve “garante” que o sistema XPTO se mantém disponível 99% do tempo (24 horas X 7 dias).
  - ›  $24 \times 7 = 168$  horas
  - › 99% de 168 corresponde a aproximadamente 166,32 horas
  - › Sua empresa garante que a **manutenção** do seu sistema ocorra em 1,68 horas?

# Desempenho

- › Preocupação com a duração de requisições feitas ao sistema
  - › Tem um alto nível de abstração, ou seja, trata do tempo de resposta do sistema e não de operações internas.
  - › Qual o tempo de resposta da operação mais importante do seu sistema?
  - › Quantas operações podem ser realizadas ao mesmo tempo, com uma duração aceitável?
- › Estímulo:
  - › Forma com que surgem os eventos:
    - › Periódica: Existe uma previsibilidade quanto ao surgimento de requisições;
    - › Estocástica: Eventos surgem de acordo com distribuição probabilística.
- › Medidas de Resposta:
  - › Duração de requisição;
  - › Prazo do processamento;
  - › Vazão (Número de operações por unidade de tempo)
  - › Instabilidade das requisições
  - › Número de operações não processadas

# Desempenho - Exemplo

- › Desejo:
  - › Espera-se que a velocidade do sistema seja o seu diferencial, de modo a contribuir com a melhora do número de vendas pela Internet.
- › Cenário de Desempenho:
  - › Ao realizar uma pesquisa o sistema deverá responder em menos de 2 segundos. Diariamente são realizadas 10.000 requisições (das 08:00 às 17:00 horas).
- › Análise do cenário:
  - › Vazão = Número de requisições / intervalo de tempo
  - › Vazão = 10.000 requisições / 540 minutos.
  - › Logo Vazão = 18,5 requisições/minuto
  - › Considerando 1 único processador, alocado em 100% para processar todas as 18,5 requisições num período de 60 segundos:
    - › Duração revisada = 60 segundos / 18,5 requisições
    - › Duração revisada = 3,25 segundos

# Testabilidade

- › Demonstra a facilidade com que o sistema pode apresentar os seus resultados de testes.
  - › Quanto tempo demora para executar os testes de regressão;
  - › Qual a cobertura dos casos de testes executados.
- › Estímulo:
  - › Início da fase de execução de testes, tanto para desenvolvimento de novo projetos, quanto para a manutenção de sistemas existentes.
- › Resposta:
  - › Percentual de declarações que foram executadas em um caso de testes;
  - › Medida para execução dos testes (Número de testes /Dia);
  - › Probabilidade de encontrar novos casos de testes

# Testabilidade - Exemplo

- › Desejo:
  - › Que o sistema seja testado no menor tempo possível
- › Cenário de Testabilidade
  - › Ao realizar qualquer tipo de manutenção no sistema, devem ser realizados os seus testes de unidade, integração e sistema em até 2 horas, com uma cobertura de 80% do código fonte e 100% das funcionalidades mais importantes do sistema.
- › Medidas de Resposta:
  - › Percentual de funcionalidades testadas;
  - › Probabilidade de falhas;
  - › Tempo para execução dos testes;
  - › Tempo para preparar o ambiente para execução dos testes.



# Trabalho em grupo

- › Projetar o cenário de qualidade - considerado como mais importante do projeto usado como tema do trabalho em grupo.
  - › Utilizar como referência:
    - › Modelo de cenário de 6 partes, estruturando o cenário em Origem, Estímulo, Ambiente, Artefato, Resposta e Medida de Resposta;
    - › Cenários genéricos de atributos de qualidade.
  - › A medida de resposta deve ser verificável para garantir a execução de testes.

# Trabalho em Grupo - Estrutura

- › Entrega completa (29/09/2012)
  - › Deve ser entregue documento impresso (Parte 1 + Parte 2)
  - › Cada grupo deverá fazer uma apresentação de 15 minutos
- › Parte 2
  - › Melhoria Proposta (\*)
    - › Basear-se nos indicadores para determinar um plano de ação, para melhoria do processo de testes
    - › O plano de ação deverá envolver a aplicação de alguma técnica de testes
    - › Determinar quais seriam os resultados esperados a partir deste plano, baseando-se em indicadores
    - › Determinar quais os riscos associados ao uso da técnica
  - › Conclusão
    - › Análise crítica do plano de ação dentro da empresa, estimando a viabilidade do plano
    - › Qual a interpretação do grupo para os resultados esperados
    - › Comparar os resultados esperados com os resultados já publicados em outras literaturas

(\*) Deve fazer uso de pelo menos 2 artigos e 1 livro de teste de software

# Referências

- › BASS, L.; CLEMENTS, P.; KAZMAN, R. **Software Architecture in Practice**. 2. ed. Boston: Addison-Wesley, 2003. 560 p. CD-ROM.
- › IEEE. **IEEE Std 1471**: recommended practice for architectural description of software-intensive systems. New York, USA: The Institute of Electrical and Electronics Engineers, 2000.
- › KLEIN, M.; KAZMAN, R. Attribute-Based Architectural Styles. **Technical Report**. Pittsburgh, Carnegie Mellon University, 1999. 82 p. Disponível em <http://www.sei.cmu.edu/reports/99tr022.pdf>. Acesso em: 20 mai. 2010.
- ›
- › KRUCHTEN, P. **The Rational Unified Process: An Introduction**. 3 ed. Boston: Addison-Wesley Professional, 2003. 336 p. CD-ROM.