

# ENGENHARIA DE REQUISITOS

Prof. Renato Matroniani



EDUCAÇÃO  
METODISTA

# EMENTA

- Introdução à engenharia de requisitos, processos de software, modelos e atividades de processos, RUP, métodos ágeis, XP, SCRUM, introdução à modelagem de sistemas.
- Práticas de Engenharia de Requisitos: Requisitos funcionais e não funcionais, documento de requisitos, especificação de requisitos, processos de engenharia de requisitos.



# OBJETIVOS

- Analisar problemas do mundo real e organizá-lo em requisitos e casos de uso para fomentar as próximas fases do desenvolvimento de software, assim como definir a quantidade de trabalho a ser realizada.



EDUCAÇÃO  
METODISTA



# Referências

- SOMMERVILLE, Ian. Engenharia de Software. 10ª ed. São Paulo: Pearson Prentice Hall, 2011. (Biblioteca Virtual) <https://plataforma.bvirtual.com.br/Leitor/Publicacao/168127/pdf/0>
- PRESSMAN, R. S. Engenharia de Software. Rio de Janeiro: McGraw Hill, 2002, 704p.
- SBROCCO, J. H.; MACEDO, P. C., Metodologias Ágeis: Engenharia de Software sob Medida., São Paulo: Érica, 2012, 256p.
- CHIKOFFSKY, Elliot. Computer-Aided Software Engineering (CASE). COMPUTER IEEE Computer Society, 1993.

# Introdução

- Os requisitos de um sistema são as descrições do que o sistema deve fazer, os serviços que oferece e as restrições a seu funcionamento.
- Esses requisitos refletem as necessidades dos clientes para um sistema que serve a uma finalidade determinada, como controlar um dispositivo, colocar um pedido ou encontrar informações.
- O processo de descobrir, analisar, documentar e verificar esses serviços e restrições é chamado Engenharia de Requisitos (RE, do inglês *requirements engineering*).

# Introdução

- O termo 'requisito' não é usado de forma consistente pela indústria de software.
- Em alguns casos, o requisito é apenas uma declaração abstrata em alto nível de um serviço que o sistema deve oferecer ou uma restrição a um sistema.
- No outro extremo, uma definição detalhada e formal de uma função do sistema.



# Introdução

- “A Engenharia de Requisitos, uma subárea da Engenharia de Software, é a etapa onde são criados e refinados os requisitos de software. Através da aplicação de técnicas e abordagens de elicitación de requisitos, os requisitos são capturados, analisados e validados com seus interessados. Por sua vez, a definição correta dos requisitos é um fator crucial para reduzir custos e riscos no projeto de software. No entanto, entende-se que a falta de um bom trabalho nesta etapa pode levar ao insucesso ou abandono do projeto.” (Rodrigo Cezario da Silva, 2011, p. iii)



# Comunicação do projeto



EDUCAÇÃO  
METODISTA



# ER sob o ponto de vista da Gestão (geral)

- Objetivos
  - Necessidade de possuir a visão geral do negócio.
  - Conhecer o cliente e seus anseios e necessidades.
- Quais os resultados esperados?
  - Identificar os *stakeholders*.
  - Identificar os diferentes pontos de vista
  - Visão geral do escopo.

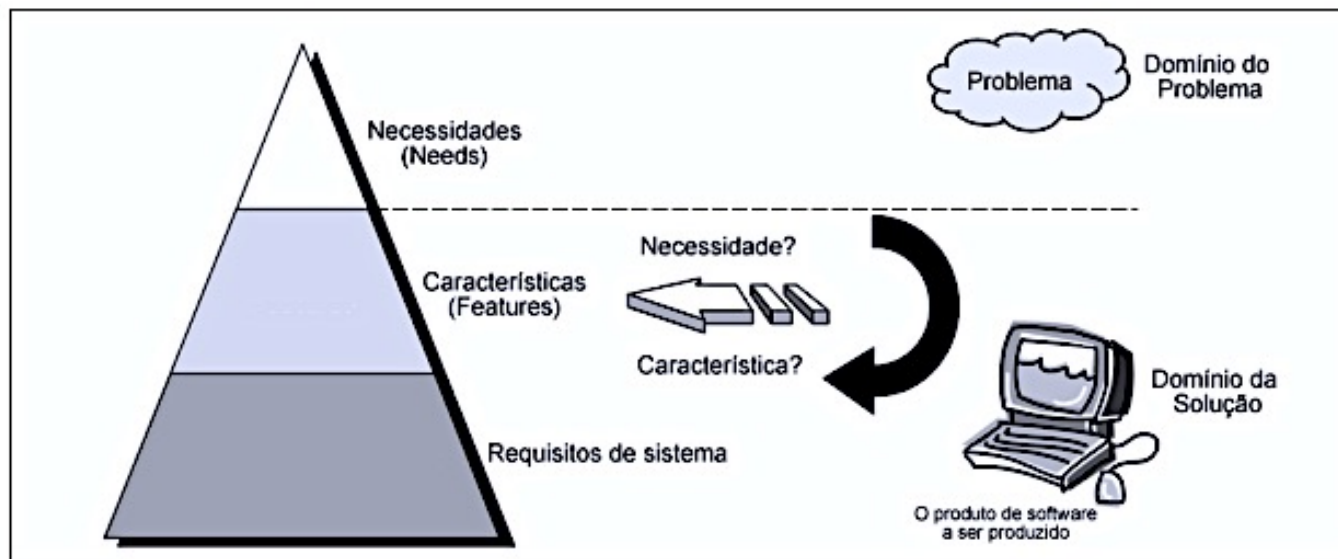


# Especificação e requisitos

- Não são a mesma coisa!
- **Especificação**: descrição detalhada das **características** de um material, uma construção, um software ou um serviço deverão **apresentar**.
- **Requisitos**: condições necessárias para se obter um certo **objetivo ou finalidade**. “Um requisito consiste na definição documentada de uma propriedade ou comportamento que um produto **deve atender**”.



# Especificação e requisitos



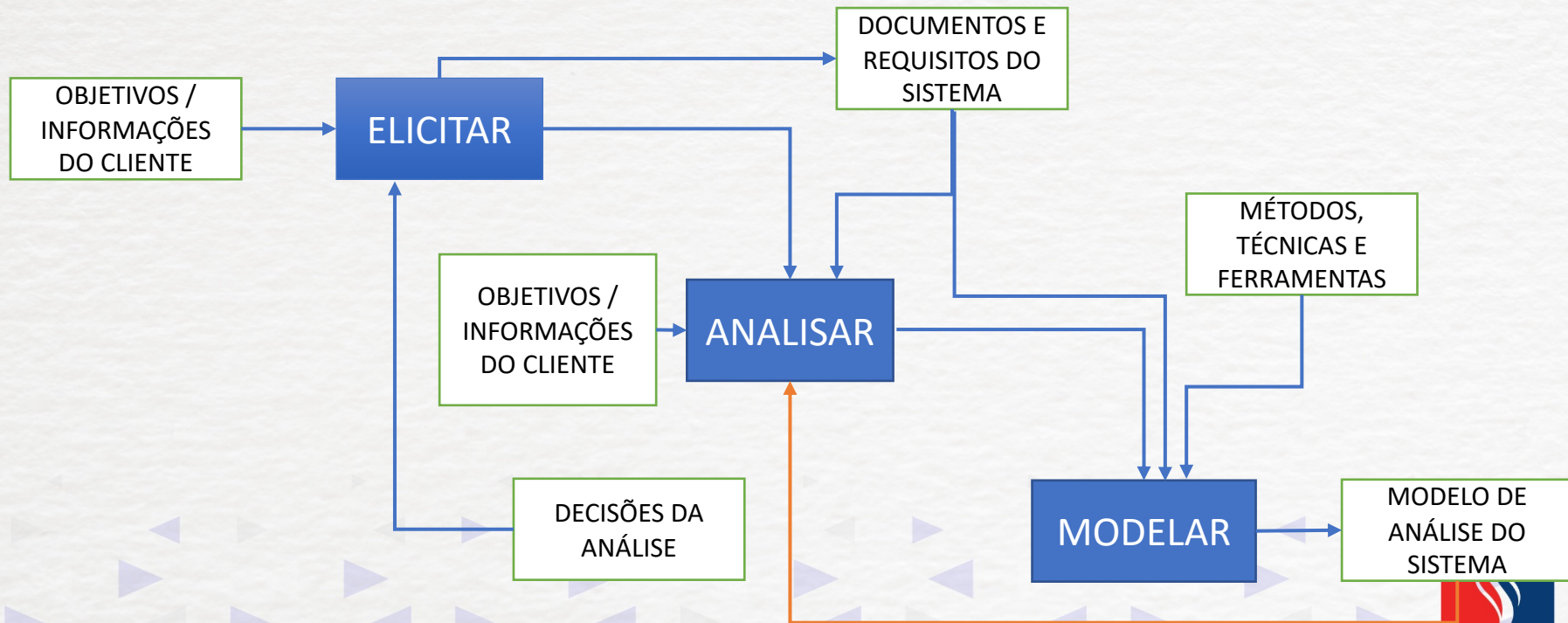
Fonte: adaptado de Leffingwell e Widrig (2003) por Rodrigo Cezario da Silva e Fabiane Barreto Vavassori Benitti



**EDUCAÇÃO  
METODISTA**



# Processo/Atividades da ER



# ELICITAR

- De forma prática, **significa obter todas as informações necessárias** para conhecer o que se quer produzir.
- De acordo com Nakagawa (2012), é composto de três atividades principais:
  - identificar fontes de informação (cliente/manuais/externas)
  - coleta de fatos (ou de dados/informações/reuniões/entrevistas...)
  - **comunicação** (cliente – engenheiro do projeto e equipe)

# ELICITAR

- Necessidade de Elicitação, segundo von Neumann, por Nakagawa (2012, adaptado): “Não há serventia em precisar de algo quando você nem sabe do que está falando”.
- Tipos de questionamento:
  - O que é desconhecido?
  - Você conhece um problema parecido?
- Para refletir: então qual a importância da elicitação?

Fonte: Nakagawa (2012)



# COLETA DE FATOS E INFORMAÇÕES

documentos

observações

entrevistas

pesquisas/  
questionários

análise dos  
protocolos

equipe  
multidisciplinar  
participativa

reuniões

reutilização de  
SW

Engenharia  
reversa do  
projeto do SW



EDUCAÇÃO  
METODISTA

Fonte: Nakagawa (2012)

# IMPORTÂNCIA DA COMUNICAÇÃO

- Fundamental para todas as fases do projeto, inclusive a fase de elicitação.
- Comunicação cliente – desenvolvedores ou Engs de Software
- Segundo Nakagawa (2012), quatro fatores devem ser considerados:
  - entendimento
  - linguagem
  - nível de abstração
  - retro-alimentação

Fonte: Nakagawa (2012)

# Verificação e Validação

- validação: “estamos construindo o produto certo?”
- verificação: “estamos construindo o produto de maneira certa?”
- Os processos de verificação e validação objetivam verificar se o software em desenvolvimento satisfaz suas especificações e oferece a funcionalidade esperada pelas pessoas que estão pagando pelo software.



# Verificação e Validação

- Esses processos de verificação iniciam-se assim que os requisitos estão disponíveis e continuam em todas as fases do processo de desenvolvimento.
- O objetivo da verificação é checar se o software atende a seus requisitos funcionais (o que será feito) e não funcionais (como serão feitos).

# Verificação e Validação

- Validação: mais geral, cujo objetivo é garantir que o SW atenda às expectativas do cliente.
- Para refletir: “a validação é necessária pois a especificação dos requisitos nem sempre refletem os desejos ou necessidades dos clientes e usuários”.

# Modelagem

- De acordo com Nakagawa (2012), a modelagem consiste na construção de modelos do sistema utilizando determinadas técnicas e métodos.
- Inicialmente é um processo abstrato.
- Consiste em três atividades:
  - representação (operações, relações, tipos)
  - organização (nível de abstração, refinamento e consistência)
  - armazenamento (classificação, indexação)

*Texto baseado em Nakagawa (2012)*



# Documentação dos requisitos

- Todo projeto, inclusive de SW, deve possuir documentação (deve seguir normas de institutos padronizadores).
- Documentação de requisitos do sistema – resultado da elicitação.
- Deve conter a especificação dos requisitos funcionais (o que será feito) e não funcionais (como será feito) – que são requisitos de qualidade de SW.
- \*esses dois tipos de requisitos serão vistos mais adiante

# Documentação dos requisitos

- Outros requisitos devem ser inclusos, como requisitos de capacidade, recursos, benefícios e aceitação.
- E sobre qualidade de software:
  - qual a relação com os requisitos?
  - existem métodos padronizados?
  - histórico de problemas de qualidade.
- PS.: ER possui relação quase que direta com Qualidade e Testes de Software (Eng de SW).

*Texto baseado em Nakagawa (2012)*

# Documentação dos requisitos – Erros comuns

- Nakagawa (2012) enuncia alguns erros comuns que ocorrem no desenvolvimento de documentação:
- ignorar parte do grupo do cliente ou um grupo de clientes.
- omitir grupo de requisitos.
- inconsistências entre grupos de requisitos.
- aceitar requisito inadequado, incorreto, impreciso, indefinido, inconsistente ou ambíguo.



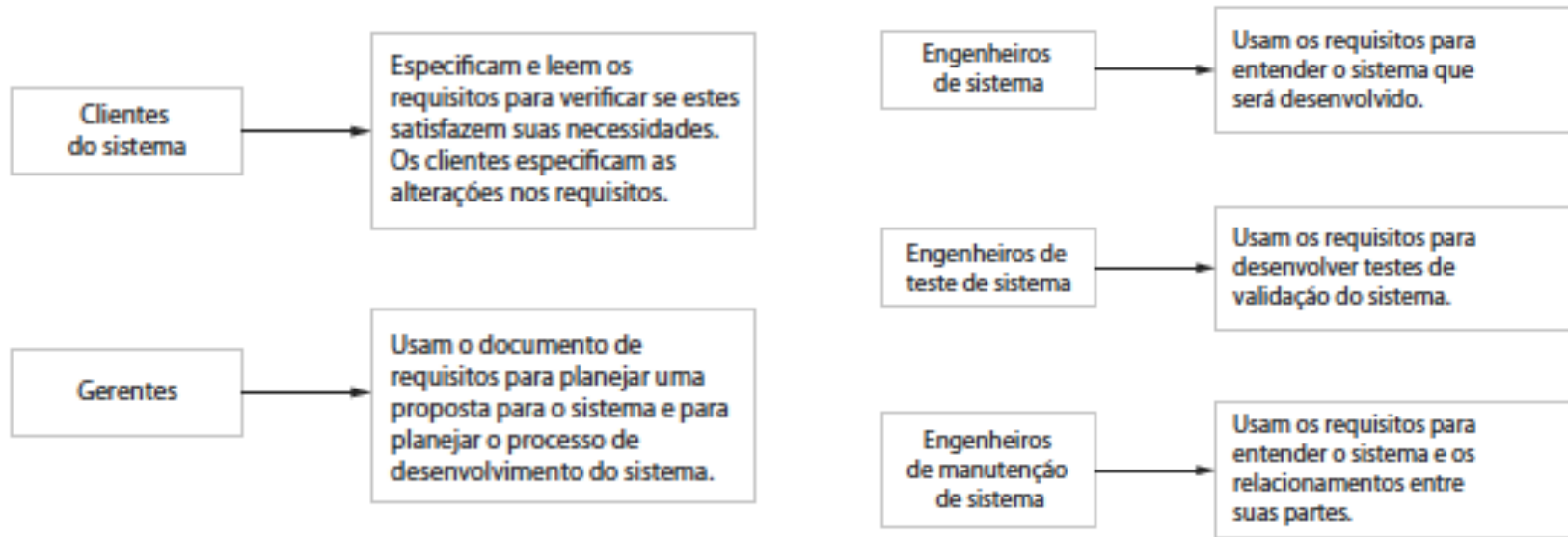
# Documentação dos requisitos

- Utilizar a **linguagem natural**
- Exemplo
  - “O sistema deve...”
- A documentação deve seguir uma sequência lógica: **entrada → processamento → saída**
- Não conter detalhes de implementação nesta etapa (exemplo: menu, arquivo).
- Deve conter um **vocabulário** com os termos a parte.
- A estrutura do documento será vista mais adiante



EDUCAÇÃO  
METODISTA

# Usuários de documentos de ER



# Sobre Qualidade de Software

- norma ISO/IEC 9126 (existem outras da ISO).
- características:

funcionalidade

confiabilidade

eficiência

manutenibilidade

usabilidade

portabilidade



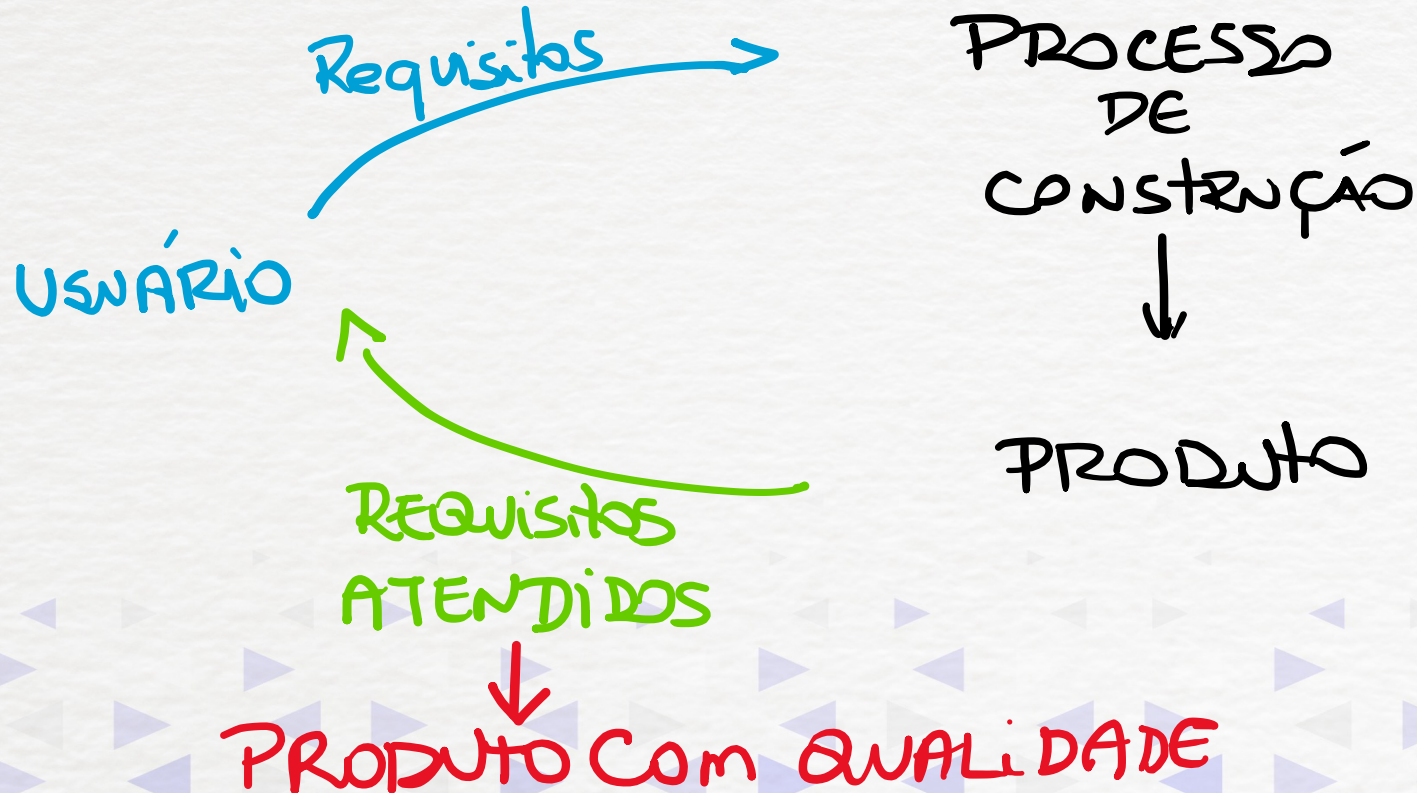
# Atributos de qualidade de SW (mais estendido)

Segurança	Compreensibilidade	Portabilidade
Proteção	Testabilidade	Usabilidade
Confiabilidade	Adaptabilidade	Reusabilidade
Resiliência	Modularidade	Eficiência
Robustez	Complexidade	Capacidade de aprendizado

*Boehm et al. (1978)*

Texto baseado em SOMMERVILLE (2011)

# Visão geral de qualidade



# Requisitos e Necessidades dos Clientes

- “Se uma empresa pretende fechar um contrato para um projeto de desenvolvimento de software de grande porte, deve definir as necessidades de forma abstrata o suficiente para que a solução para essas necessidades não seja predefinida.
- Os requisitos precisam ser escritos de modo que vários contratantes possam concorrer pelo contrato e oferecer diferentes maneiras de atender às necessidades da organização do cliente.
- Uma vez que o contrato tenha sido adjudicado [pertencente a alguém], o contratante deve escrever para o cliente uma definição mais detalhada do sistema, para que este entenda e possa validar o que o software fará.
- Ambos os documentos podem ser chamados documentos de requisitos para o sistema”.



# Requisitos de usuário e Requisitos de Sistema

- Requisitos de usuário são declarações, em uma linguagem natural com diagramas, de quais serviços o sistema dever fornecer a seus usuários e as restrições com as quais este deve operar.
- Requisitos de sistema são descrições mais detalhadas das funções, serviços e restrições operacionais do sistema de software.
- O documento de requisitos do sistema (às vezes, chamado especificação funcional) deve definir exatamente o que deve ser implementado.
- Pode ser parte do contrato entre o comprador do sistema e os desenvolvedores de software.

# Requisitos de usuário e Requisitos de Sistema

- Um requisito de usuário pode ser expandido para diversos requisitos de sistema.
- Requisitos de usuários – mais gerais
- Requisitos de sistema – mais específicos
- Detalhamento é fundamental para os leitores de diversos níveis.
- Leitor dos requisitos de usuário geralmente não estão interessados nos detalhes do sistema.



# Requisitos de usuário e Requisitos de Sistema

- Requisitos com abordagem ágil será tratado mais adiante.
- Para grandes projetos a fase de elicitação não utiliza metodologia ágil.



**EDUCAÇÃO  
METODISTA**

*Texto de SOMMERVILLE (2011)*



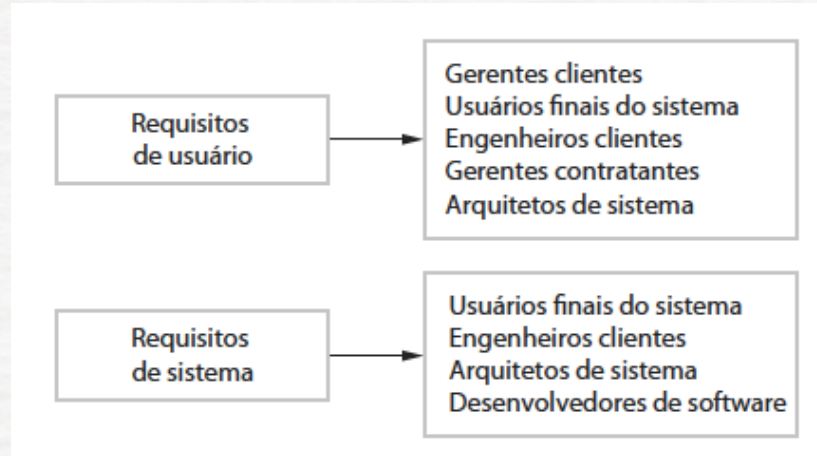
# Requisitos de usuário e Requisitos de Sistema

## Definição de requisitos de usuário

1. O MHC-PMS deve gerar relatórios gerenciais mensais que mostrem o custo dos medicamentos prescritos por cada clínica durante aquele mês.

## Especificação de requisitos de sistema

- 1.1 No último dia útil de cada mês deve ser gerado um resumo dos medicamentos prescritos, seus custos e as prescrições de cada clínica.
- 1.2 Após 17:30h do último dia útil do mês, o sistema deve gerar automaticamente o relatório para impressão.
- 1.3 Um relatório será criado para cada clínica, listando os nomes dos medicamentos, o número total de prescrições, o número de doses prescritas e o custo total dos medicamentos prescritos.
- 1.4 Se os medicamentos estão disponíveis em diferentes unidades de dosagem (por exemplo, 10 mg, 20 mg), devem ser criados relatórios separados para cada unidade.
- 1.5 O acesso aos relatórios de custos deve ser restrito a usuários autorizados por uma lista de controle de gerenciamento de acesso.



Fonte: SOMMERVILLE (2011)

# Requisitos de usuário e Requisitos de Sistema

