

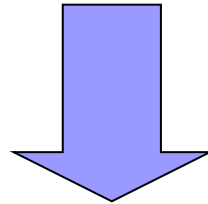


# Modelagem de Software

**Prof. Fernando Maia da Mota**

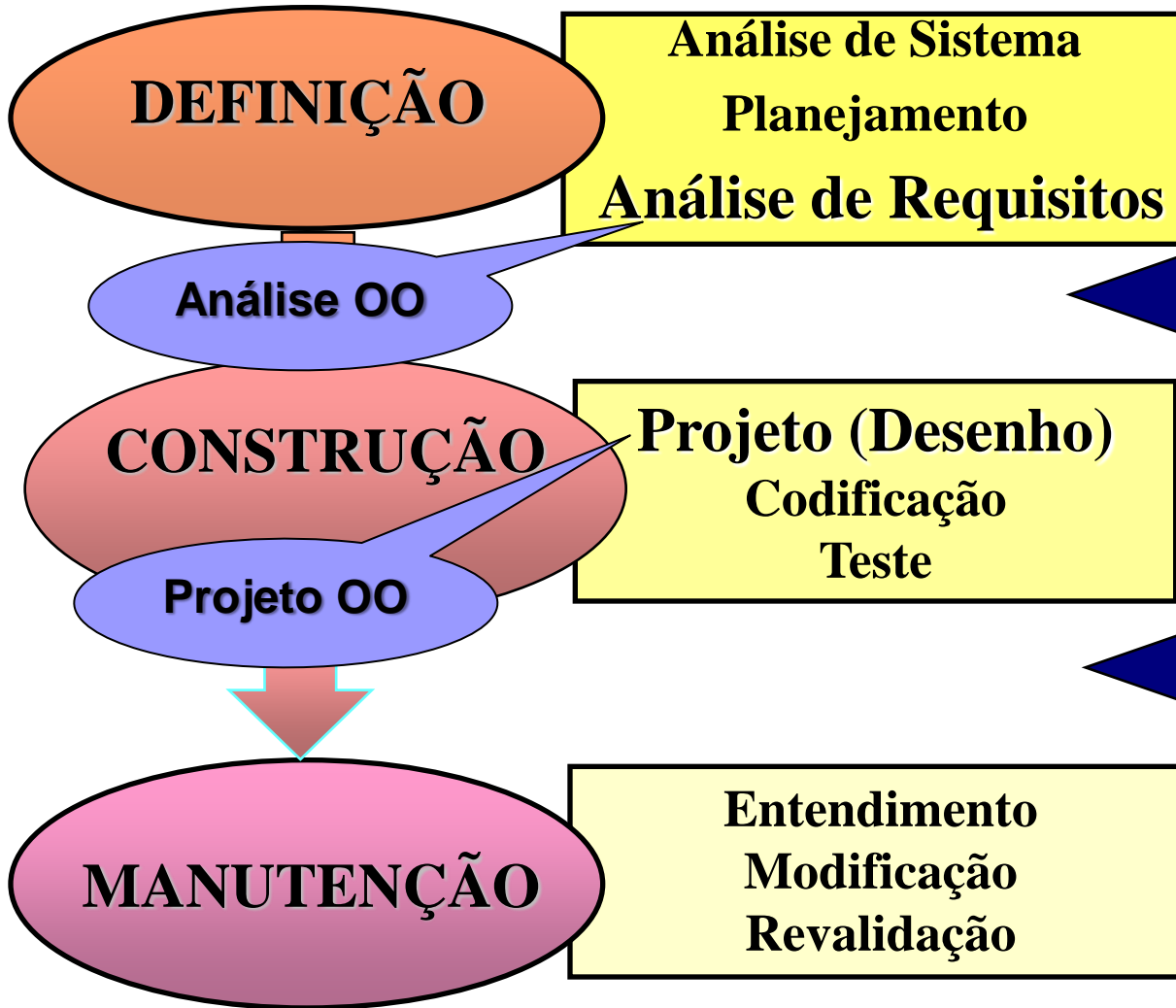
Slides gentilmente cedidos por Profa. Dra. Maria Istela Cagnin Machado  
UFMS/FACOM

***Um software de boa qualidade só pode ser obtido caso esteja em conformidade aos requisitos***



**O bom entendimento dos requisitos é a base para a *Análise e o Projeto de Software***

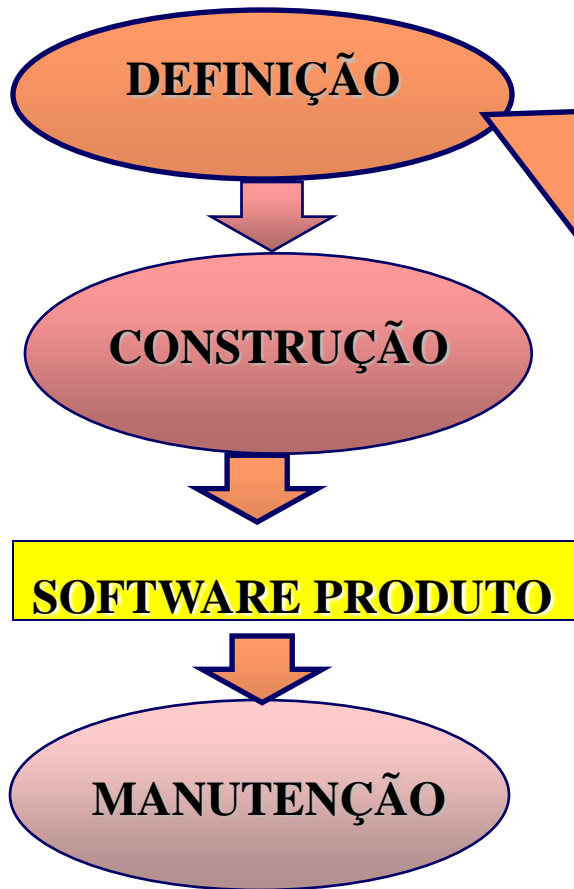
# Fases dos Modelos de Processo de Software



## ATIVIDADES DE APOIO

- *Controle e Acompanhamento do Projeto de Software*
- *Revisões Técnicas Formais*
- *Garantia de Qualidade de Software*
- *Gerenciamento de Configuração de Software*
- *Preparação e Produção de Documentos*
- *Gerenciamento de Reusabilidade*
- *Medidas*
- *Gerenciamento de Riscos*

# Requisitos de Software



*“o que”*

- informações a serem processadas
- funções e desempenho desejados
- interfaces que devem ser estabelecidas
- restrições do projeto
- critérios de validação requeridos



# Análise de Requisitos

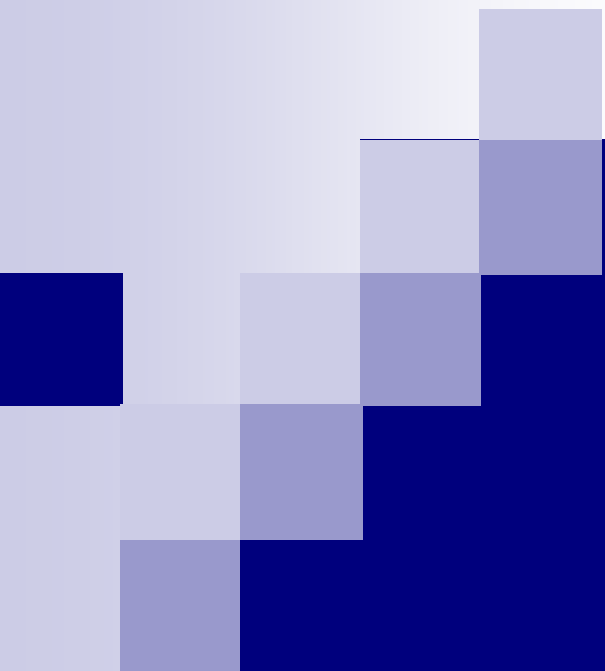
- Clarear + extrair requisitos
- Descobrir, tornar explícito, obter o máximo de informação para o conhecimento do problema em questão

# ***Exemplo de um documento de requisitos***



***Vamos falar nesta disciplina  
sobre Análise e Projeto de  
Software!***





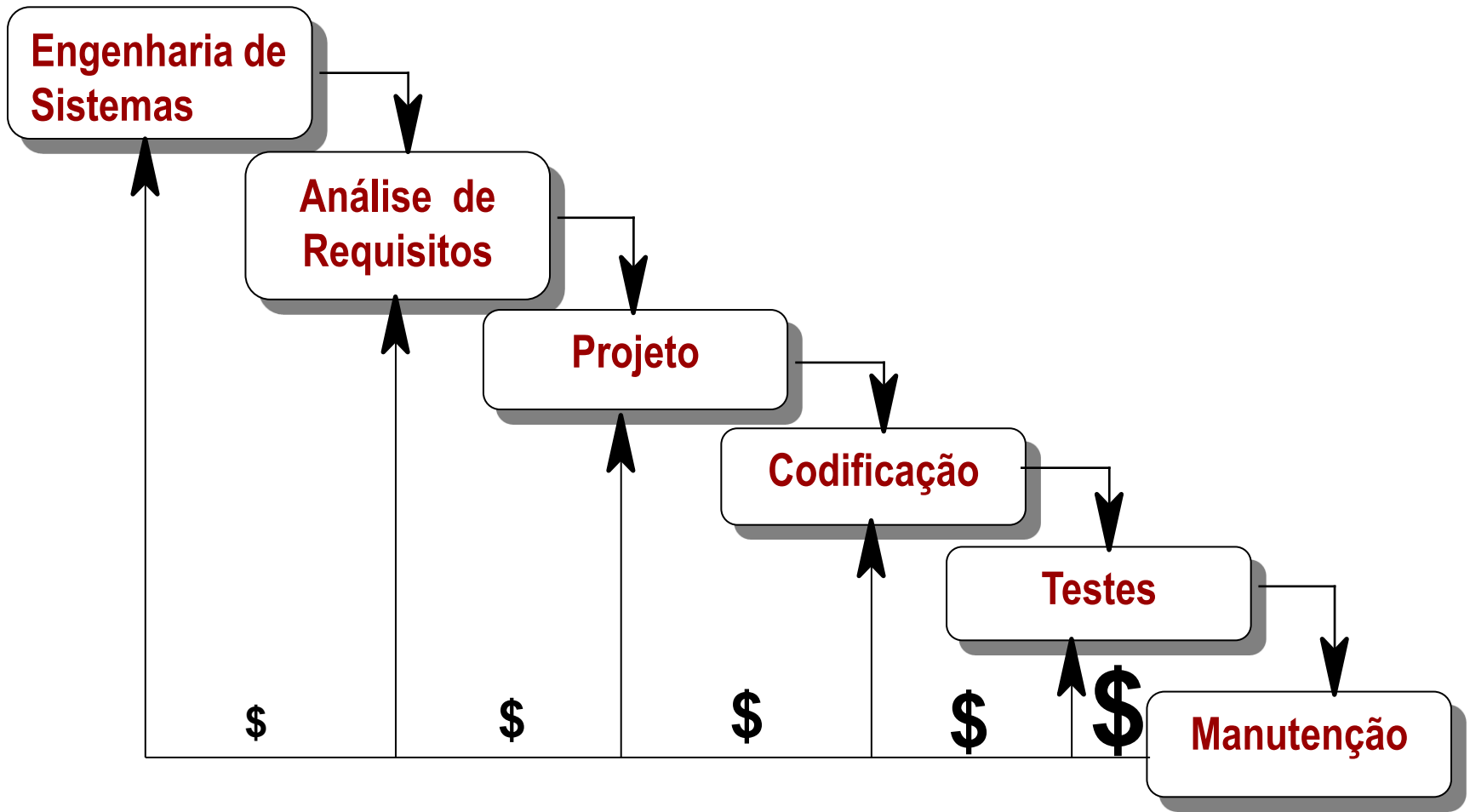
Para que servem os  
modelos?





# A Modelagem como Técnica de Análise e Projeto

- “Um modelo é uma abstração de alguma coisa, cujo propósito é permitir que se conheça essa coisa antes de construí-la” (*Rumbaugh et al.*)
- Um modelo omite os detalhes não essenciais e, portanto, sua manipulação é mais fácil do que a da entidade original



# A Modelagem como Técnica de Análise e Projeto

- A abstração é uma fundamental capacidade humana que nos permite lidar com elementos complexos
- Engenheiros, artistas e artesãos constroem modelos há milhares de anos para testar projetos antes de executá-los

# A Modelagem como Técnica de Análise e Projeto

- No desenvolvimento de sistemas abstrair significa concentrar-se no que um objeto é e faz, antes de decidir **como** ele deve ser implementado
- Na construção de um modelo não se deve procurar a verdade absoluta, e sim a adequação a algum propósito. Não há um único modelo “correto” de uma situação, apenas modelos adequados e inadequados

# A Modelagem como Técnica de Análise e Projeto

- O uso da abstração durante a análise significa:
  - lidar apenas com conceitos do domínio da aplicação
  - não ter que tomar decisões sobre o projeto e a implementação antes do problema ser compreendido

# A Modelagem como Técnica de Análise e Projeto

- Os modelos servem para diversos objetivos:
  - Entender o que está sendo construído
    - obter visualização do delineamento das ideias
  - Melhorar comunicação entre os membros da equipe e com os clientes
  - Testar uma entidade antes de lhe dar forma
  - Reduzir a complexidade

# Prática de Modelagem de Software

- Em Engenharia de Software, duas categorias principais de modelos são criadas:
  - modelos de análise
  - modelos de projeto



# Prática de Modelagem de Software

- **Modelos de análise**: representam os requisitos do cliente mostrando o software em três domínios diferentes: o domínio de *informação*, o domínio funcional e o domínio comportamental
- **Modelos de projeto**: representam características de software que ajudam os profissionais a construí-lo efetivamente: a arquitetura, a interface do usuário e os componentes

# Princípios da Modelagem (Análise de Software)

- 1- O domínio de informação de um problema precisa ser representado e entendido
- 2- As funções a serem desenvolvidas pelo software devem ser definidas
- 3- O comportamento do software (como consequência de eventos externos) precisa ser representado

# Princípios da Modelagem (Análise de Software)

- 4- A tarefa de análise deve começar na informação essencial e partir para informações mais detalhadas

# Princípios da Modelagem (Projeto de Software)

- 1- O projeto deve estar relacionado ao modelo de análise
- 2- Sempre considere a arquitetura do sistema a ser construído
- 3- O projeto dos dados é tão importante quanto o projeto de funções de processamento
- 4- As interfaces (tanto externas quanto internas) precisam ser projetadas com cuidado

# Princípios da Modelagem (Projeto de Software)

- 5- O projeto de interface do usuário deve estar sintonizado com as necessidades do usuário final
- 6- O projeto em nível de componente deve ser funcionalmente independente
- 7- Os componentes devem ser fracamente acoplados uns aos outros e ao ambiente externo

# Princípios da Modelagem (Projeto de Software)

- 8- Modelos de projeto devem ser facilmente compreensíveis
- 9- O projeto deve ser desenvolvido iterativamente



# Prática de Modelagem de Software

- Existem muitos modelos diferentes que podem ser desenvolvidos na análise de sistemas



# Prática de Modelagem de Software

- Diferentes tipos de sistemas podem exigir modelos diferentes para realçar as características importantes
- Diferentes projetos podem exigir diferentes ferramentas de modelagem face aos padrões de documentação impostos por organizações externas



# Prática de Modelagem de Software

- Qualquer ferramenta deve ter as seguintes características:
  - deve ser gráfica, com adequado detalhamento textual de apoio
  - deve permitir que o sistema seja visualizado de forma subdividida
  - deve ter mínima redundância
  - deve ajudar o leitor a prognosticar o comportamento do sistema
  - deve ser transparente para o leitor

# Trabalho Prático (05/03 às 23h no email [mota.fernandomaia@gmail.com](mailto:mota.fernandomaia@gmail.com))

- Indicar os membros do grupo, o nome e uma breve descrição do sistema que será desenvolvido na disciplina.
- Máximo de 03 pessoas por grupo.

# Trabalho Prático (12/03 até às 18:30 no email [mota.fernandomaia@gmail.com](mailto:mota.fernandomaia@gmail.com) e trazer impresso na aula)

- Elaborar o documento de requisitos do sistema do projeto
  - Utilize como base o documento de requisitos do sistema de clínica veterinária

# Bibliografia

- SANCHES, ROSELY. Material Didático: Engenharia de Software. ICMC-USP, 2004.
- PRESSMAN, ROGER S. Engenharia de Software. 6ª edição. Rio de Janeiro: McGraw-Hill, 2006.
- M. Blaha, J. Rumbaugh. Modelagem de Projetos baseados em Objetos com UML 2. Editora Campus, 2006.