



UNIVERSIDADE FEDERAL DO PIAUÍ – UFPI  
CAMPUS SENADOR HELVÍDIO NUNES DE BARROS - CSHNB  
CURSO DE BACHARELADO EM SISTEMAS DE INFORMAÇÃO



# **Engenharia de Software I**

## **Engenharia de Requisitos – Parte 2**

**Professora Pâmela Carvalho**

**31/01/23**

# **Introdução a Engenharia de Requisitos**

# Desafios da análise de requisitos

- Como **descobrir** os requisitos;
- Como **comunicar** os requisitos para as outras fases ou equipes do projeto;
- Como **lembrar** dos requisitos durante o desenvolvimento e verificar se foram todos atendidos ;
- Como **gerenciar** a mudança.



# Validação dos requisitos

- Será que realmente entendi o que o cliente deseja?
- Devo me certificar de que não houve falha em nossa interação (comunicação).
- Há diversas técnicas de validação:
  - Revisões de requisitos
  - Prototipação
  - Geração de casos de testes
  - Etc.

Características	Definição
Correto	É correto se, e somente se, cada requisito expresso for encontrado também no software.
Não ambíguo	É não ambíguo se, e somente se, cada requisito declarado seja suscetível a apenas uma interpretação.
Completo	É completo se, e somente se, conter toda e apenas a informação necessária para que o software correspondente seja produzido.
Consistente	É consistente se, e somente se, nenhum dos requisitos do documento, tomado individualmente, está em conflito com qualquer outro requisito do mesmo documento.
Classificado por importância / estabilidade	Se existe indicações no documento quanto à importância ou estabilidade do requisito.
Verificável	É verificável se, e somente se, para cada um dos requisitos contidos no documento, existe um processo finito e economicamente viável através do qual uma pessoa ou máquina possa assegurar que o produto de software atende ao requisito.
Modificável	É modificável se, e somente se, modificações possam ser agregadas ao documento de forma fácil, completa e consistente, com relação a sua estrutura e estilo.
Rastreável	É rastreável se, e somente se, a origem de cada um de seus requisitos é clara e a referência a cada um deles é facilitada nos documentos subsequentes do processo ou em uma melhoria da documentação do sistema.

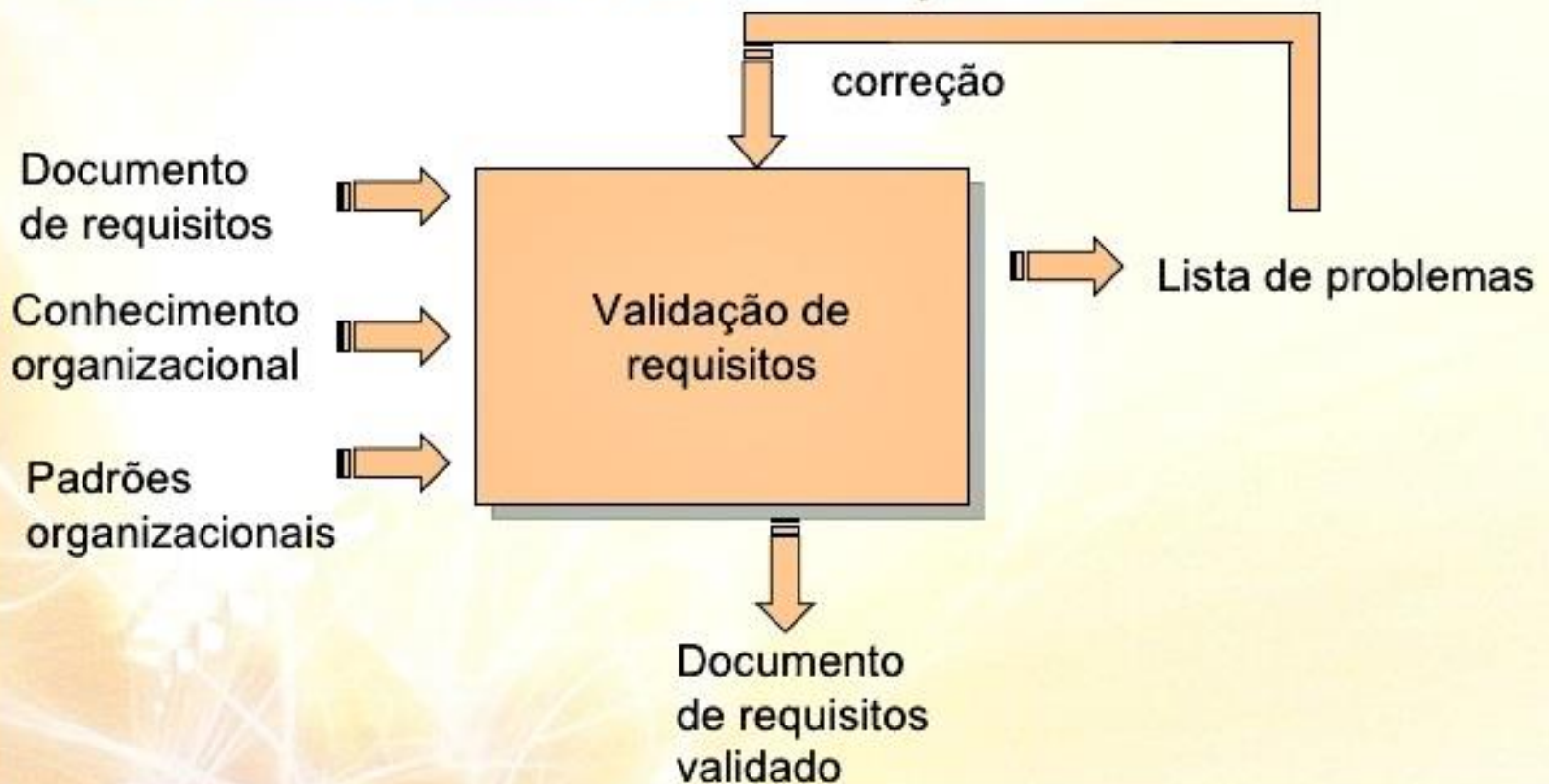
# Técnicas de validação de requisitos

- **Revisão de requisitos:**
  - Analise manual sistemática dos requisitos
- **Prototipação:**
  - Uso de um modelo executável do sistema para checar os requisitos.
- **Geração de casos de teste:**
  - Desenvolver testes para validar a implementação dos requisitos.
- **Análise automatizada da consistência:**
  - Uso de ferramenta para verificar a consistência do modelo.



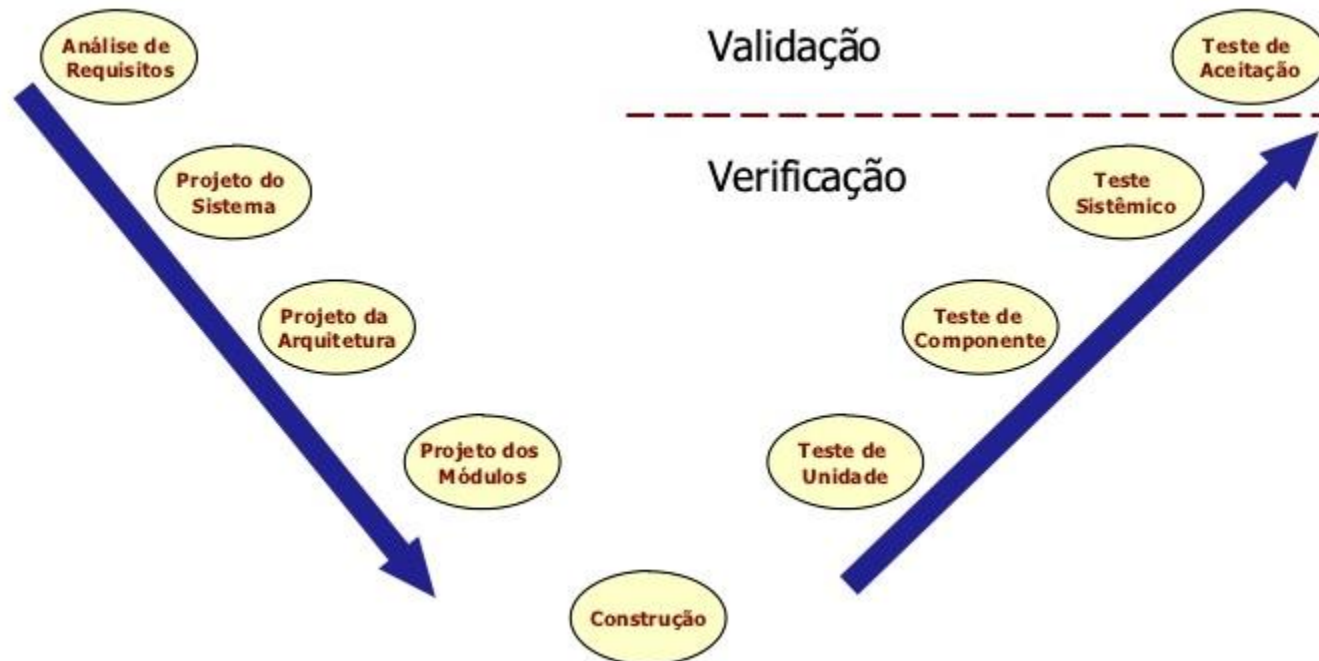
# Técnicas de validação de requisitos

- Entradas e saídas da validação

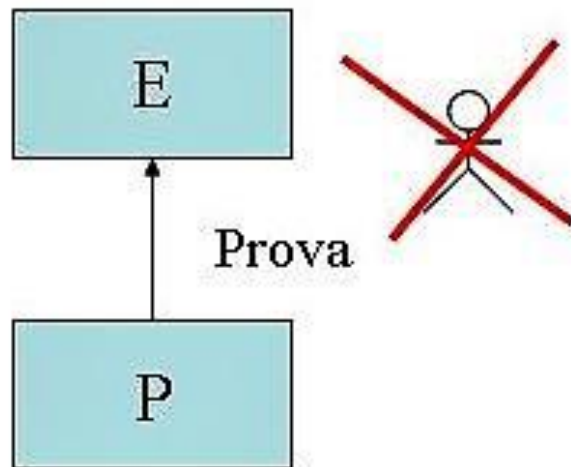


# Validação dos requisitos

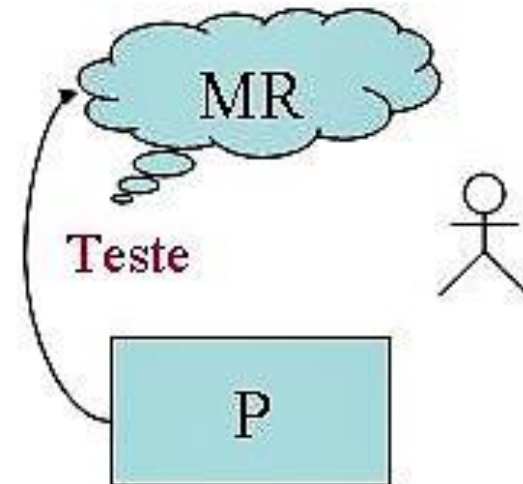
- A validação deve demonstrar que os **requisitos definem o sistema que o cliente realmente deseja**.
- Custos com erros de requisitos são altos.
  - Consertar um erro de requisitos após entrega do sistema pode custar mais de 100 vezes o custo de um erro de implementação.



# Verificação X Validação



P implementa E?  
*Estamos Construindo o  
Produto da Maneira Correta?*



P atende as expectativas do MR?  
*Estamos Construindo o  
Produto Desejado?*

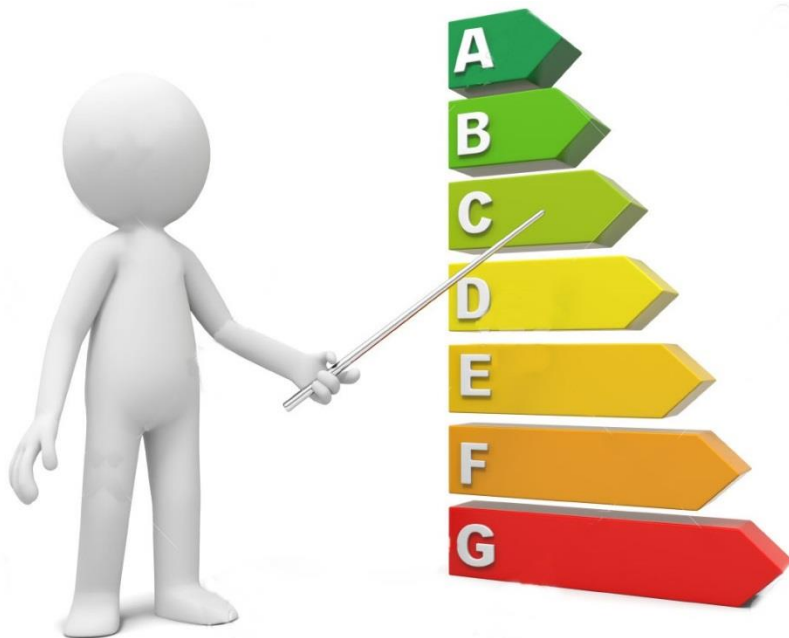


# **Classificação dos Requisitos**

# Classificação

- Esta etapa consiste basicamente em **agrupar os diversos requisitos coletados em categorias bem-definidas.**

Por exemplo:



## **REQUISITOS FUNCIONAIS:**

Descrevem o **comportamento do sistema**, suas ações para cada entrada, ou seja, é aquilo que descreve o que tem que ser feito pelo sistema.

## **REQUISITOS NÃO FUNCIONAIS:**

Expressam **como deve ser feito**. Em geral, se relacionam com padrões de qualidade como confiabilidade, performance, robustez, etc.

# Requisitos funcionais e não-funcionais

- O requisito é uma condição cuja exigência deve ser satisfeita. O mesmo deve ser especificado no início do projeto.

Uma especificação pode ser um **documento escrito**, um modelo gráfico, um modelo matemático formal, uma coleção de cenários, um protótipo, ou qualquer combinação entre eles.

- Se a condição é **produzir algo**, diz-se que o requisito é **funcional**.
- Se a condição é **caracterizar algo** (propriedade, comportamento, restrição, etc,...), diz-se que o requisito é **não-funcional**.

# Requisitos funcionais e não-funcionais

## Requisitos funcionais

Correspondem à listagem de todas as coisas que o sistema deve fazer;

## Requisitos não funcionais

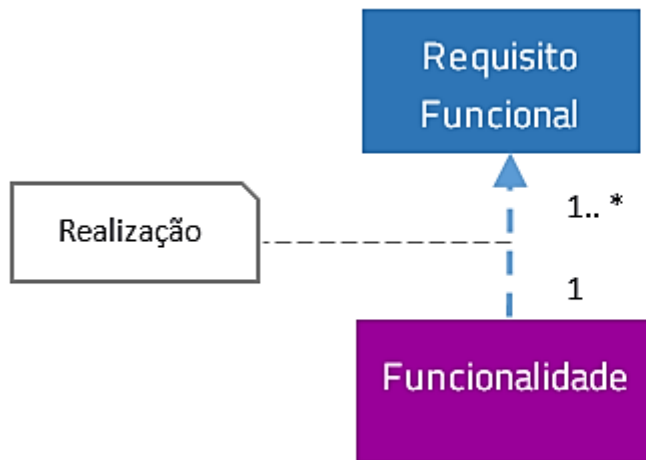
São restrições e qualidades que se coloca sobre como o sistema deve realizar seus requisitos funcionais;



- Existe uma definição propagada na literatura de Engenharia de Software que afirma que um **Requisito Funcional** define **o que** o sistema fará, e o **Requisito Não-Funcional** define **como** o sistema fará.

# Requisito Funcional

- É necessário entender que **uma funcionalidade não necessariamente realizará apenas um Requisito Funcional**.
- Uma funcionalidade pode realizar **vários** Requisitos Funcionais.



Para entender melhor isso vamos a um exemplo mais básico. Imaginemos um sistema que possui uma tela para “Manutenção de Clientes”, que mantém os dados cadastrais de um cliente no sistema.

Estamos falando de uma **única funcionalidade**. Nesta tela é possível incluir/alterar/consultar/excluir clientes dos tipos pessoa física e pessoa jurídica.

- Mas quantos requisitos podem ser realizados (atendidos) por esta funcionalidade?



# Requisito Funcional

Nesse exemplo, podem ser atendidos até **8 requisitos**.

- Vejamos a lista a seguir:

Requisitos Funcionais (Identificador e Nome)
RF001 – Incluir cliente pessoa física
RF002 – Alterar cliente pessoa física
RF003 – Consultar cliente pessoa física
RF004 – Excluir cliente pessoa física
RF005 – Incluir cliente pessoa jurídica
RF006 – Alterar cliente pessoa jurídica
RF007 – Consultar cliente pessoa jurídica
RF008 – Excluir cliente pessoa jurídica

Uma **funcionalidade** pode realizar um ou mais Requisitos Funcionais.

**Requisito funcional** não é uma funcionalidade, é uma necessidade funcional (uma função) que o software deve atender.

Uma **funcionalidade** será executada por um ator (um ator sistêmico [pelo próprio sistema] ou um ator humano [usuário final]).

# Requisito Funcional

- Um Requisito Funcional de qualidade precisa atender alguns atributos específicos, como por exemplo:

Atributo	Referente a
Unidade	O RF deve propor uma única coisa apenas. Não deve atender a mais de uma exigência. O RF "Incluir cliente" não é unitário, pois se refere a incluir clientes de tipos diferentes (pessoa física e jurídica), assumindo assim várias responsabilidades, quando deveria assumir apenas uma.
Compleitude	O RF deve ser autocontido, deve ter "início/meio/fim", ser completo. O RF "Pagar fatura" não é completo, só conta "parte da estória". Para ser completo deveria ser algo como "Pagar fatura com cartão de crédito para cliente pessoa física".
Consistência	O RF não deve contradizer outro RF do mesmo escopo do projeto. É como termos dois RFs se propondo a fazer uma mesma coisa, mas cada RF se propondo a fazer esta coisa de uma forma diferente.
Atomicidade	Um RF para ser atômico precisa também ter unidade, pois atomicidade remete a assumir apenas uma responsabilidade. Mas também deve ser algo indivisível, não podendo ser decomposto. Muitos RFs possuem conjunção, dependem de outros para se realizarem. Onde temos dois RFs "Realizar compra de produto" e "Realizar pagamento com cartão de crédito" na realidade, se pensarmos em atomicidade, temos um único RF que é "Realizar compra de produto com pagamento em cartão de crédito".
Não-Ambiguidade	Um RF não pode ser ambíguo, não pode propor algo que não fica claro o que é. O RF "Emitir relatório" não quer dizer nada. Relatório de que, para que? "Emitir relatório de saldo" já é melhor, mas ainda é ruim. Saldo de que? Seria não ambíguo se não deixasse dúvidas, algo como "Emitir relatório de saldo da conta corrente do cliente pessoa física".
Verificável	Não adianta ter um RF se ele não é palpável, possível de associar com um artefato de construção, de testes. Um RF tem que ser testável, tem que ser possível atestar que o RF foi atendido, foi construído, foi homologado. Para isso tem que ser também rastreável.
Rastreável	Deve ser possível achar o RF no sistema pronto, funcional e executável. Como saber se um RF foi atendido? Para isso é necessário ter rastreabilidade, e isso só é possível ligando as pontas (associar o RF à interface gráfica, que será associada a um caso de uso, que será associado a funcionalidades, que serão implementadas etc.).
Prioridade	Um RF Essencial é algo muito diferente de um RF Desejável, possuem valores para o negócio completamente diferentes. O RF deve possuir sua prioridade, isso interfere diretamente no projeto do software.

# Requisito Não-Funcional (RNF)

## O que é um Requisito Não-Funcional?

- Como o próprio nome diz, é uma “não funcionalidade”, ou seja, trata-se de algo que não é uma funcionalidade, mas que precisa ser realizado para que o software atenda seu propósito.

- Definem propriedades e restrições do sistema
  - Exemplos: segurança, desempenho, espaço em disco
- Podem ser do sistema todo ou de partes do sistema
- Requisitos não-funcionais podem ser mais críticos que requisitos funcionais
  - Se não satisfaz, o sistema é inútil

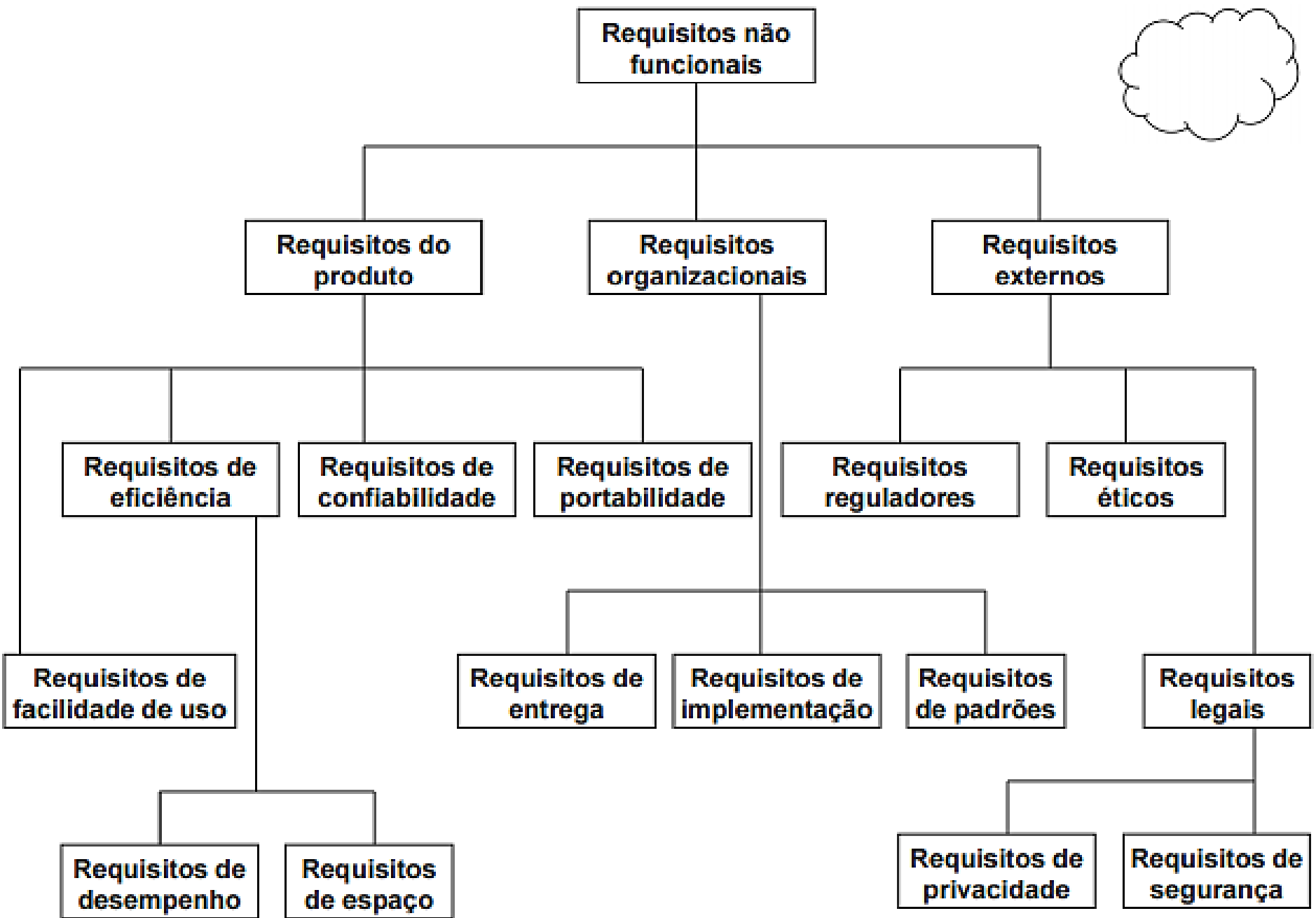


# Requisito Não-Funcional (RNF)

- **Requisitos do Produto**
  - Especificam o comportamento do software (ex.: desempenho)
- **Requisitos Organizacionais**
  - Consequência de políticas e procedimentos das empresas (ex.: padrões do cliente)
- **Requisitos Externos**
  - Derivados do ambiente ou fatores externos ao sistema (ex.: legislação)



- **Requisitos do Produto**
  - A interface do usuário deve ser implementada como simples HTML
- **Requisitos Organizacionais**
  - Todos os documentos entregues devem seguir o padrão de relatórios XYZ-00
- **Requisitos Externos**
  - Informações pessoais dos usuários não podem ser vistas pelos operadores do sistema





# Requisito Não-Funcional (RNF)

- Verificação de RNF:

- Às vezes são de difícil verificação
- Idealmente, requisitos não-funcionais devem ser mensuráveis
  - Após a implementação, estes podem ser testados objetivamente



## Métricas de RNF:

- Velocidade
- Facilidade de uso
- Confiabilidade
- Robustez

# Requisito Não-Funcional (RNF)

- **Métricas:**

- **Velocidade**

- Transações processadas por segundo
- Tempo de resposta
- Tempo de atualização de tela

- **Facilidade de uso**

- Tempo gasto em treinamento
- Número de *frames* de ajuda



- **Confiabilidade**

- Tempo médio para falhar
- Probabilidade de indisponibilidade
- Taxa de ocorrência de falhas
- Disponibilidade

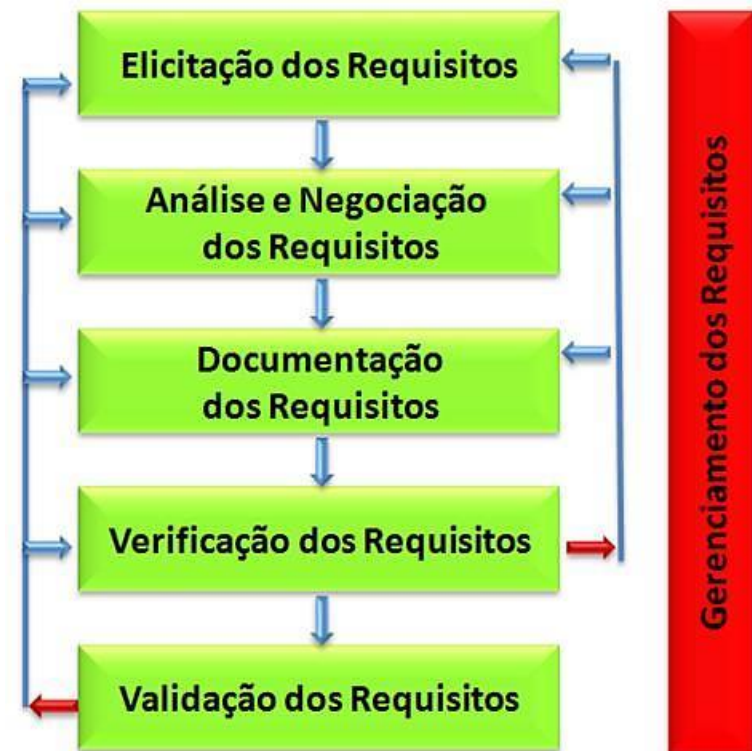
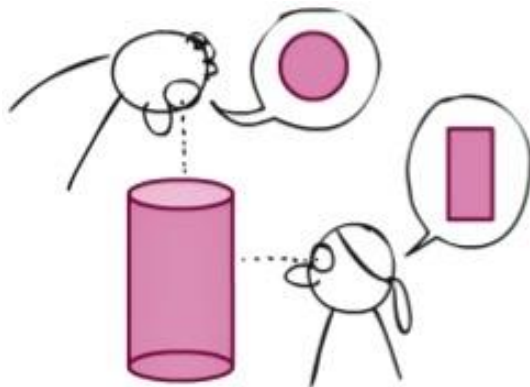
- **Robustez**

- Tempo de reinício após uma falha
- Porcentagem de eventos que causam falhas

# Modelos de documentos de especificação de requisitos

- O resultado final da análise e especificação de requisitos e das outras atividades da fase de definição devem ser apresentados aos clientes para que eles possam validá-lo.

Este documento oferece a concordância entre clientes, analistas e desenvolvedores sobre o que deve ser desenvolvido.



# Modelos de documentos de especificação de requisitos

- Cada desenvolvedor utiliza um modelo específico para elaborar este documento.
- A sua estrutura muitas vezes depende do método que está sendo utilizado.



- Este modelo deve ser basicamente textual, utilizando o mínimo de termos técnicos, e ilustrados como modelos gráficos que demonstrem mais claramente a visão que os analistas tiveram dos problemas e dos requisitos para o futuro sistema.

# Estrutura de um documento de requisitos

- 1. Introdução
- 2. Definição dos Requisitos do Usuário
- 3. Especificação dos Requisitos do Sistema
- 4. Arquitetura do Sistema
- 5. Modelos do Sistema
- 6. Evolução do Sistema
- 7. Apêndices
- 8. Índice





# Documento de requisitos

- Fonte: IEEE/ANSI (830-1998)
- **1. Introdução**
  - 1.1 Propósito do documento
  - 1.2 Escopo do sistema
  - 1.3 Glossário, acrônimos e abreviaturas
  - 1.4 Referências
  - 1.5 Descrição do resto do documento



# Documento de requisitos

- Fonte: IEEE/ANSI (830-1998)
- **2. Descrição geral**
  - 2.1 Perspectiva do produto
  - 2.2 Funções do produto
  - 2.3 Características dos usuários
  - 2.4 Restrições gerais
  - 2.5 dependências



# Documento de requisitos

- Fonte: IEEE/ANSI (830-1998)
- **3. Requisitos específicos**
  - requisitos funcionais, não-funcionais, etc:
    - funcionalidade, interfaces externas, desempenho, restrições, atributos do sistema, caract. qualidade, ...



# Documento de requisitos

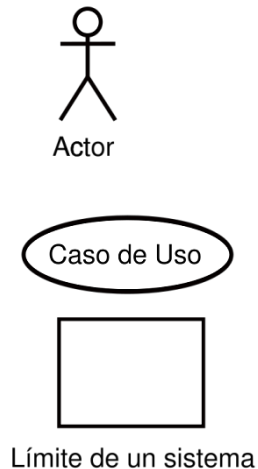


- **4. Arquitetura do Sistema**
- **5. Modelos do Sistema**
  - Diagrama de Atores
  - Modelo de Caso de Uso
  - Modelo de Análise
  - Modelo de Projeto
  - Diagrama de Pacotes
- **6. Evolução do Sistema (Futuro)**
- **7. Apêndices**
- **8. Índice**

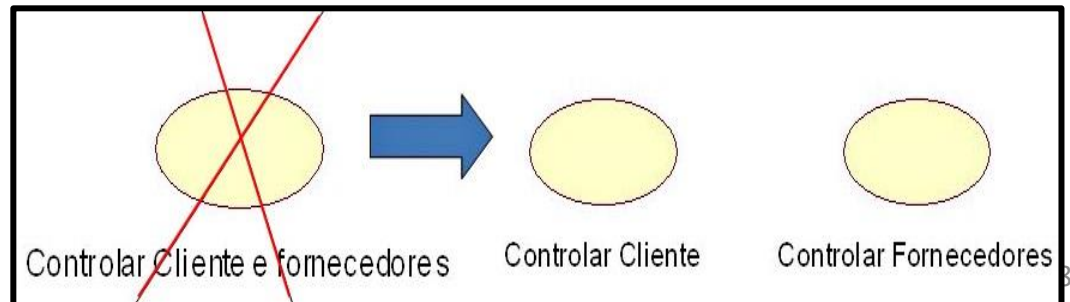
# Especificando requisitos utilizando casos de uso

- Após saber quais as tarefas associadas a cada papel de usuário, é hora de elaborar os **casos de uso (use cases)** que permitem definir as funções de aplicação que o sistema deverá oferecer para o usuário.

Os casos de uso podem ser utilizadas durante a análise e especificação dos requisitos para descrever a funcionalidade do sistema.



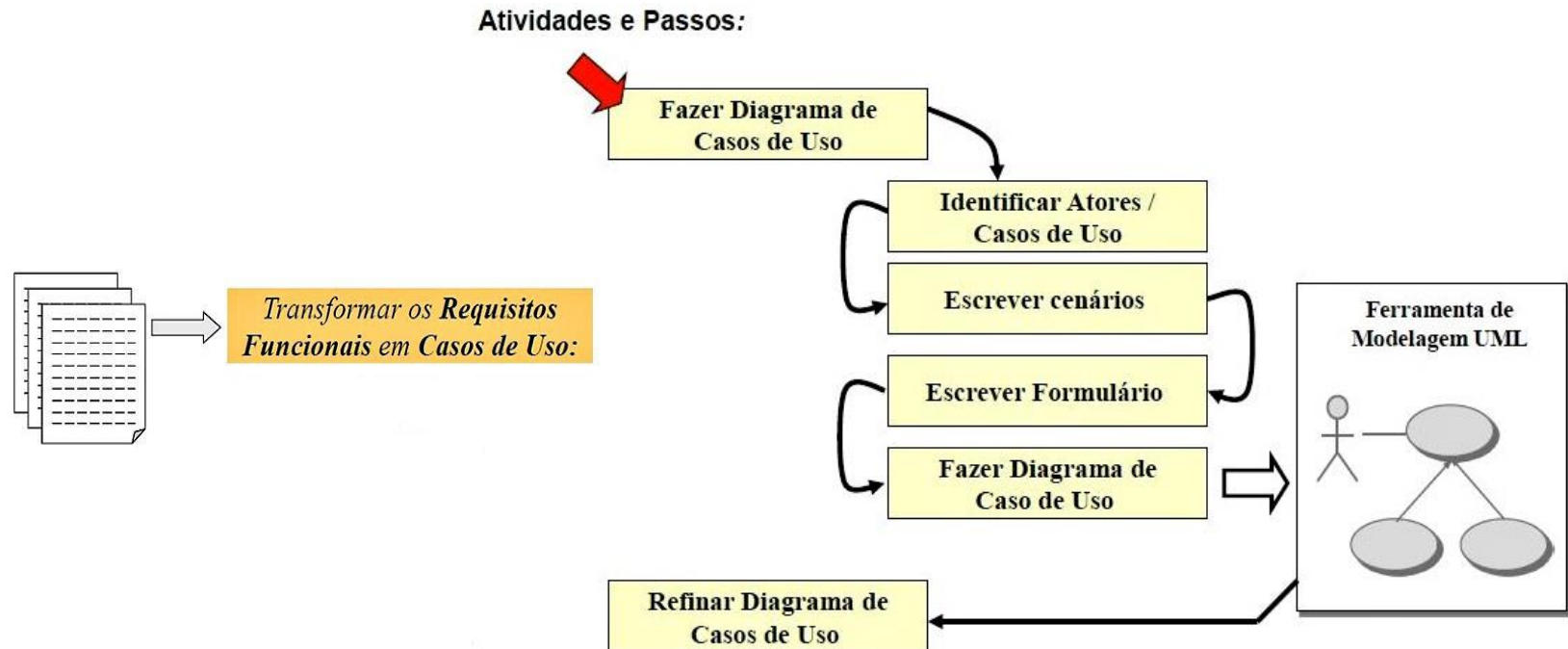
- O caso de uso modela o diálogo entre um ator e o sistema;
- Não se deve representar para o mesmo Caso de Uso mais do que uma funcionalidade.





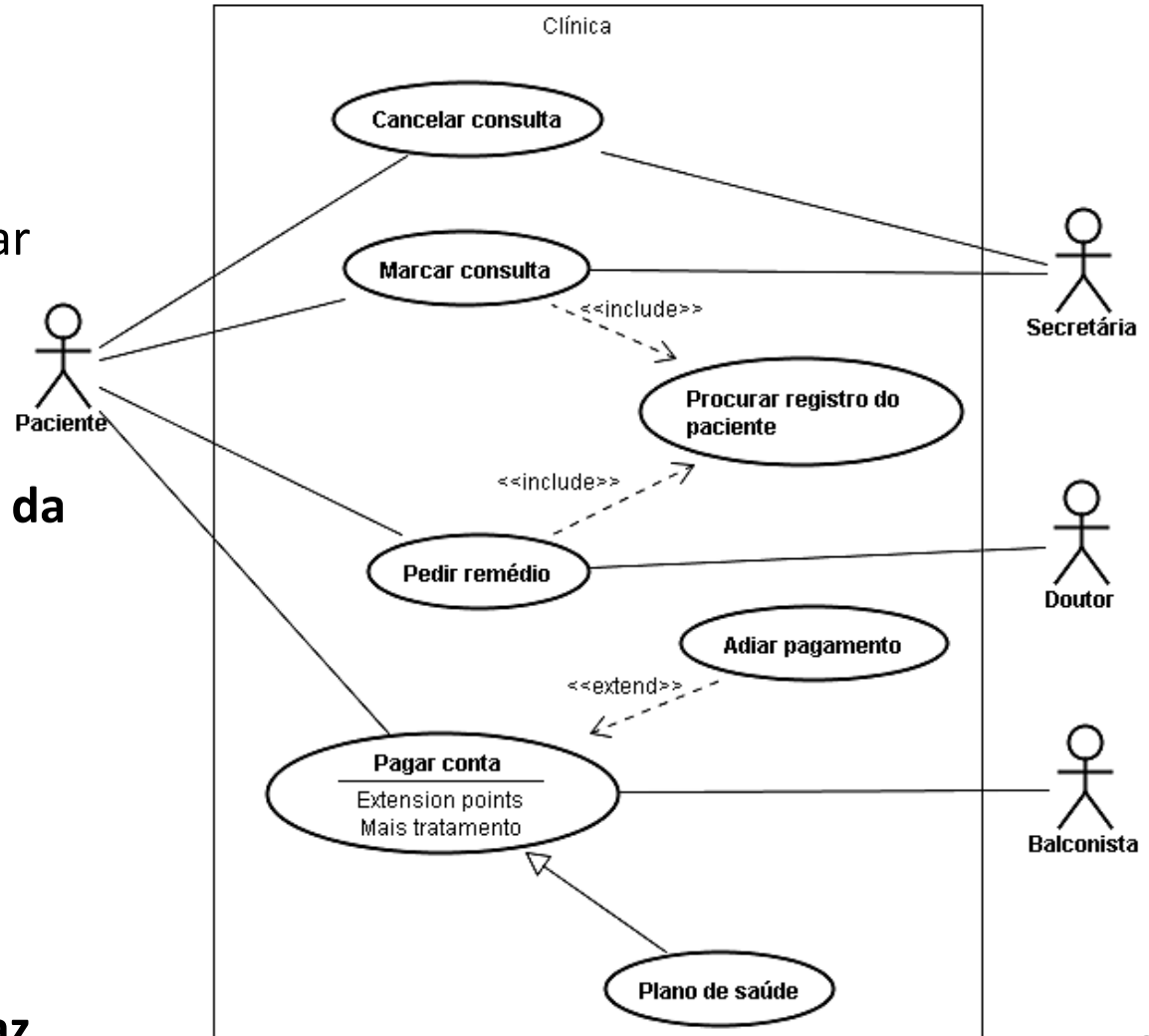
# Especificando requisitos utilizando casos de uso

- Cada caso de uso define um **requisito funcional** do sistema.
  - **Ex.** Em um sistema bancário: consulta de saldo, empréstimos e saques.
- Ao definir os requisitos funcionais, o caso de uso define o comportamento, as respostas esperadas e os casos de testes que devem validar a implementação do sistema.



# Especificando requisitos utilizando casos de uso

- Um caso de uso **especifica o comportamento do sistema** sem especificar como este será implementado.
- Os **comportamentos descrevem as funções da aplicação** que caracterizam a funcionalidade do sistema.
- Um caso de uso **representa o que o sistema faz e não como o sistema faz.**



# Abreviações e Glossário

Abreviação	Significado	Explicação / Condição ou situação no sistema
A	Administrador	Usuário com maiores privilégios no sistema
AT	Auto-treinamento	Um dos três perfis de avaliação. O operador/treinando solicita ao sistema uma avaliação que lhe é montada de modo randômico a partir de alguns parâmetros
CT	Certificação Técnica	Um dos três perfis de avaliação. Os supervisores (RL/RS) agendam com antecedência dia e hora da avaliação. É o teste que certifica o treinando/operador.
O	Operador	Usuário. Treinando que realiza as avaliações.
RL	Responsável Local	Usuário. Responsável, na unidade da empresa, por um grupo de operadores. Propõe, elimina e valida questões e avaliações.
RS	Responsável Setorial	Usuário. Responsável por um setor da empresa. Coordena um ou mais RL. Propõe, elimina e valida questões e avaliações.
TO	Treinamento Orientado	Um dos três perfis de avaliação. Serve para os RS/RL diagnosticarem o estágio da aprendizagem dos operadores.
V	Validador	Usuário. Checa e valida as questões propostas pelos RS/RL.
M	Módulo	Refere-se aos módulos do sistema.
	Backup	Refere-se à cópia de dados de um dispositivo para o outro com o objetivo de posteriormente os recuperar (os dados), caso haja algum problema.
	Logon	É a ação necessária para acessar um sistema computacional restrito inserindo uma identificação, podendo esta ser ou não única para cada usuário, e a senha relacionada a ela. Uma vez logado, o usuário passa a ser identificado no sistema, sendo restringido ou permitido a acessar recursos do sistema.

# Gerenciamento de requisitos

- Gerenciamento de requisitos é o processo de **controlar as mudanças dos requisitos** durante:
  - O processo da engenharia de requisitos
  - O desenvolvimento do sistema



- Requisitos são **inevitavelmente incompletos e inconsistentes**
  - **Requisitos novos** surgem durante o processo de acordo com **mudanças nas necessidades** do negócio e um entendimento melhor do sistema é desenvolvido.
  - **Diferentes pontos de vista têm diferentes requisitos** e esses geralmente são contraditórios.

Amanhã (01/02) haverá atividade para  
compor a 2ª nota

**Elaboração de Questões (1,0)**

**Até a próxima aula...**