

# Engenharia de Software I

Prof. Orlando Saraiva Júnior  
[orlando.nascimento@fatec.sp.gov.br](mailto:orlando.nascimento@fatec.sp.gov.br)



Como o cliente explicou...



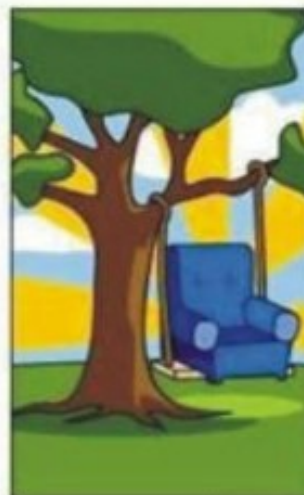
Como o líder de projeto entendeu...



Como o analista projetou...



Como o programador construiu...



Como o consultor de negócios descreveu...



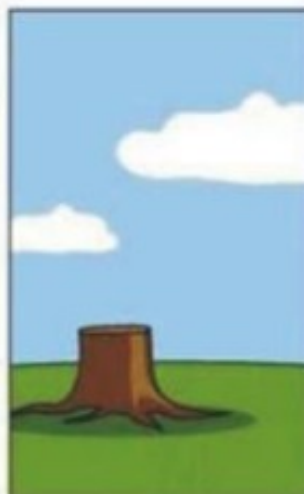
Como o projeto foi documentado...



Que funcionalidades foram instaladas...



Como o cliente foi cobrado...



Como foi mantido...



O que o cliente realmente queria...

# **Estudo de Viabilidade**

O objetivo de um estudo de viabilidade é avaliar sob o ponto de vista operacional, técnico, econômico e organizacional se o projeto é viável.

O documento destina-se aos *stakeholders* do sistema ( Sommerville, 2011 p. 70)

Visa:

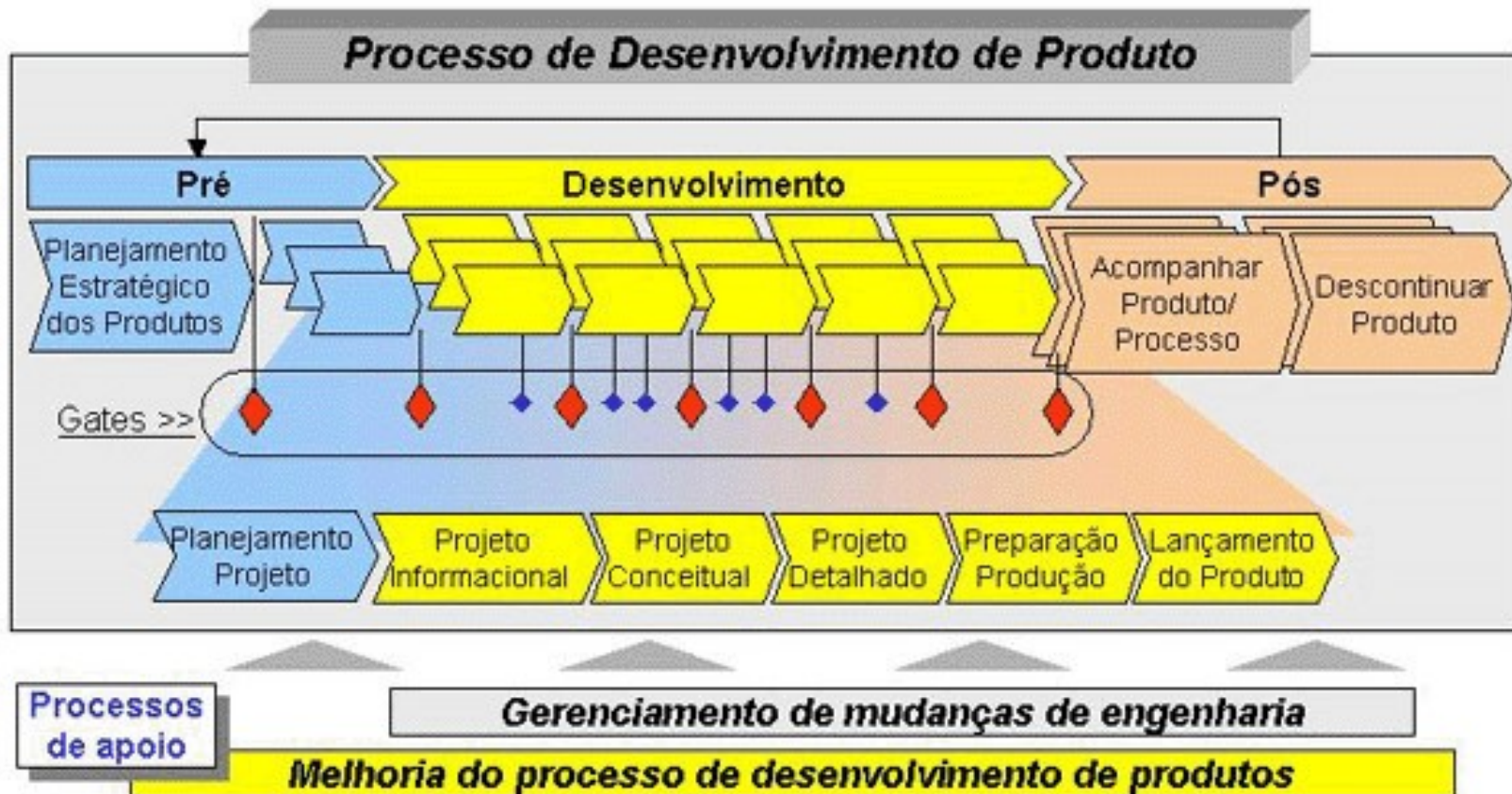
Tanto a tomada de decisão

Como a sugestão de possíveis alternativas

Deve oferecer informações para ajudar na decisão

- Se o projeto pode ou não ser feito
- Se o produto final irá ou não beneficiar os usuários interessados
- Escolha das alternativas entre as possíveis soluções
- Há alternativas melhores ?

# EV - Fase pré desenvolvimento



Fonte: Rozenfeld

## Sistema organizacional apresentado

- Usuários, políticas, funções, objetivos, etc..

## Problemas com o sistema apresentado

- Inconsistências, funcionalidades inadequadas, performance

## Objetivos e outros requisitos para o novo sistema

- O que precisa mudar ?

## Restrições

- Incluindo requisitos funcionais e não funcionais

## Alternativas possíveis

## Vantagens e desvantagens



## Operacional

- Medida do grau de adequação da solução para a organização

## Técnica

- Avaliação da praticidade da solução técnica específica e a disponibilidade dos recursos técnicos e dos especialistas

## Cronograma

- Avaliação do cronograma

## Econômica

- Avaliação do custo/benefício de um projeto ou solução.

Avaliação da urgência do problema ou aceitação da solução

Há dois aspectos de viabilidade operacional a serem considerados:

- O problema vale a pena ser resolvido ?
- Como os stakeholders sentem-se em relação ao problema ?

A solução ou a tecnologia proposta é prática ?

Já possuímos a tecnologia necessária ?

Já possuímos o conhecimento técnico necessário ?

# Viabilidade de cronograma

---

Dado nosso conhecimento técnico, os prazos do projeto são razoáveis ?

Alguns projetos são iniciados com prazos específicos.

O escopo é viável no cronograma proposto ?

Baseado nos requisitos o analista pode considerar a relação custo/benefício para cada alternativa.

Avaliar nesta etapa os custos:

- Custo de desenvolvimento do sistema
- Custo de instalação
- Custos operacionais
  - Manutenção

Diversos riscos:

- Tecnologia;
- Pessoas;
- Organização;
- Ferramentas;
- Requisitos;
- Estimativas;
- Complexidade;

- Ausência de recursos importantes no projeto;
- Mudança de requisitos que impactem diretamente nos marcos do projeto;
- Mudanças que requer retrabalho;
- Tamanho do software subestimado;
- O cliente não sabe o que quer;
- Reestruturação da organização durante a execução do projeto;
- O cliente fica indisponível;

Considerar os diferentes tipos de riscos envolvendo:

- Rotatividade de pessoal
- Atrasos na especificação
- Não prever necessidade do projeto
- Projetos paralelos que interfiram no desenvolvimento do projeto
- Dentre outros...



Após o esforço inicial, discutido anteriormente, deve-se elaborar um relatório de viabilidade :

Para cada aspecto apresentado, deve haver uma seção de avaliação

Qual a melhor alternativa

# **Levantamento de Requisitos**

# Processo de engenharia de requisito

---

Os processos de engenharia de requisitos podem incluir quatro atividades de alto nível.

Elas visam avaliar se o sistema é útil para a empresa (**estudo de viabilidade**), descobrindo requisitos (**elicitação e análise**), convertendo-os em alguma forma-padrão (**especificação**), e verificar se os requisitos realmente definem o sistema que o cliente quer (**validação**).

# O que é um Requisitos

---



Condição ou capacidade que um sistema ( ou seu componente) deve alcançar ou possuir para satisfazer um contrato, padrão, especificação ou outro documento formalmente imposto

# Levantamento de Requisitos

---

Sommerville (2003) propõe um processo genérico de levantamento e análise que contém as seguintes atividades:

**Compreensão do domínio:** Os analistas devem desenvolver sua compreensão do domínio da aplicação;

**Coleta de requisitos:** É o processo de interagir com os stakeholders do sistema para descobrir seus requisitos. A compreensão do domínio se desenvolve mais durante essa atividade;

**Classificação:** Essa atividade considera o conjunto não estruturado dos requisitos e os organiza em grupos coerentes;

# Levantamento de Requisitos

---

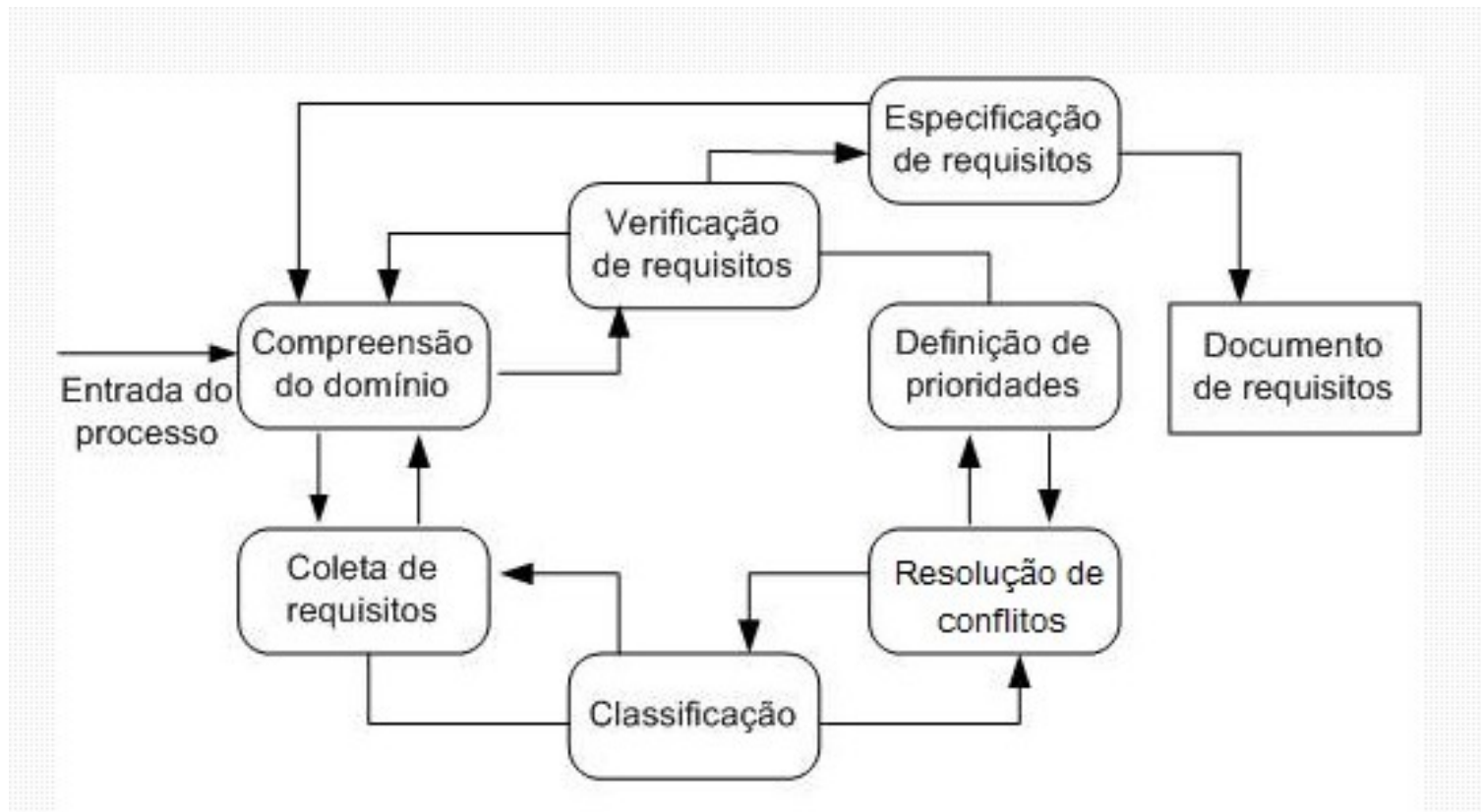
**Resolução de conflitos:** Quando múltiplos stakeholders estão envolvidos, os requisitos apresentarão conflitos. Essa atividade tem por objetivo solucionar esses conflitos;

**Definição das prioridades:** Em qualquer conjunto de requisitos, alguns serão mais importantes do que outros. Esse estágio envolve interação com os stakeholders para a definição dos requisitos mais importantes;

**Verificação de requisitos:** Os requisitos são verificados para descobrir se estão completos e consistentes e se estão em concordância com o que os stakeholders desejam do sistema.

# Levantamento de Requisitos

A análise de requisitos é um processo iterativo:



## **Requisitos funcionais**

Declarações de serviços que o sistema deve fornecer, como o sistema deve reagir a entradas específicas e como deve se comportar em determinadas situações.

## **Requisitos não funcionais**

Restrições sobre os serviços ou as funções oferecidas pelo sistema.



Descrevem a funcionalidade ou os serviços do sistema.

Dependem do tipo de software, das expectativas dos usuários e do tipo de sistema que está sendo desenvolvido.

Requisitos funcionais do usuário são descritos de forma bem geral, mas os requisitos funcionais de sistema descrevem a função de sistema detalhadamente.

# Requisitos Funcionais (exemplo)

---

[RF001] – O sistema deve permitir que os professores realizem lançamento de notas e falta dos alunos.

[RF002] – O sistema deve permitir que os professores editem o lançamento de notas e falta dos alunos.

[RF003] – O sistema deve permitir emissão de histórico escolar.

[RF004] – O sistema deve calcular as médias de cada disciplina e fornecer a situação ( aprovado/ reprovado / em recuperação).

# Requisitos não funcionais

---

Definem as propriedades de sistemas e restrições

- Confiabilidade, tempo de resposta e espaço em disco. Restrições: capacidade dos dispositivos de E/S, representações de dados, etc.

Requisitos não funcionais dizem respeito ao sistema como um todo. Alguns podem restringir o processo que é utilizado para desenvolver o sistema.

Podem ser mais críticos que requisitos funcionais. A falha em atender um requisito não funcional de sistema pode inutilizar o sistema.

# Classificação Requisitos não funcionais

---

## **Requisitos de produto**

Requisitos que especificam o comportamento do produto. Ex: velocidade de execução, confiabilidade, portabilidade, facilidade de uso, etc..

## **Requisitos organizacionais**

Requisitos que são consequência de políticas de procedimentos nas organizações do cliente e do desenvolvedor. Ex: padrões de processos que devem ser utilizados, requisitos de implementação, etc.

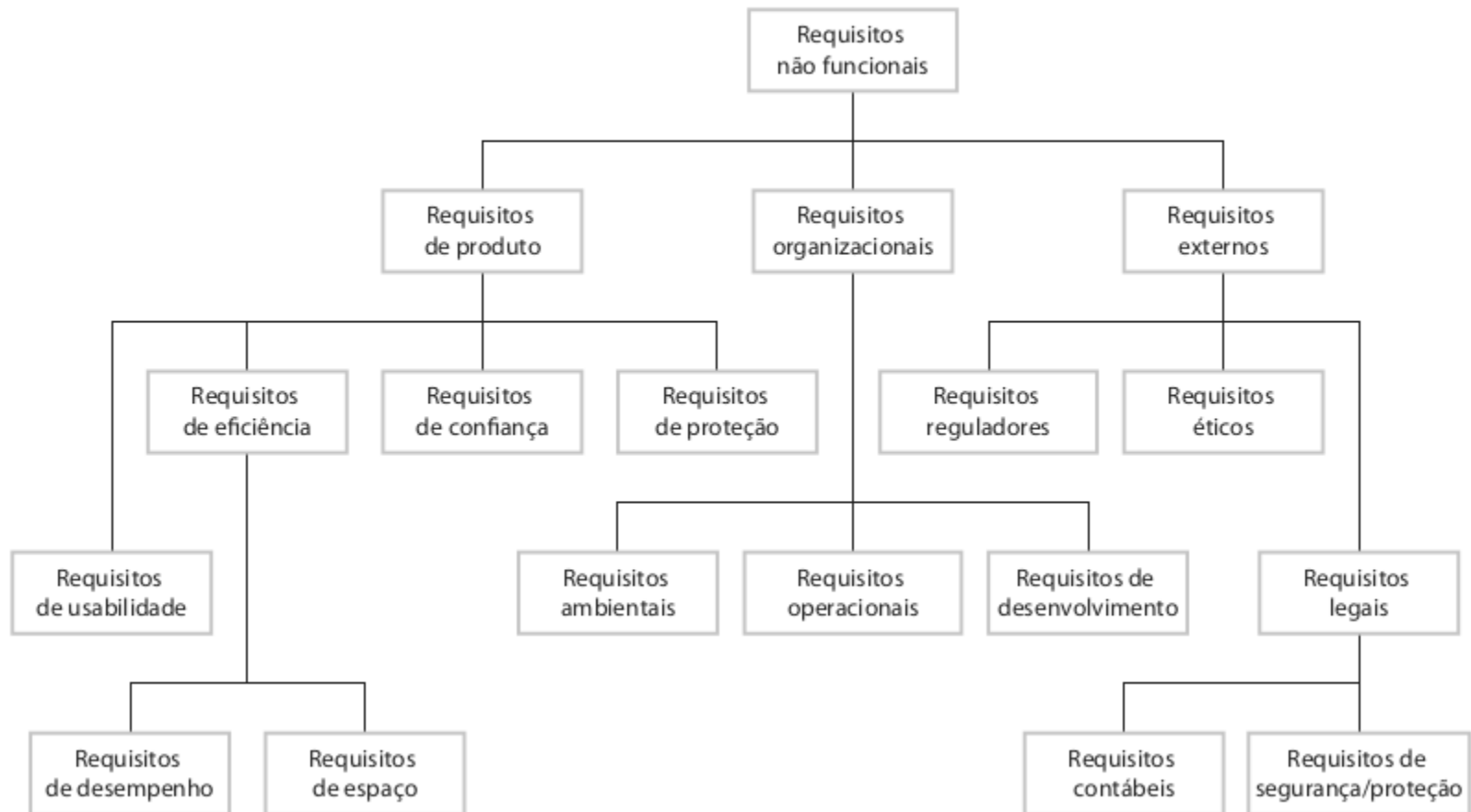
## **Requisitos externos**

Requisitos procedentes de fatores externos ao sistema e a seu processo de desenvolvimento. Ex: requisitos de interoperabilidade, requisitos legais e os requisitos éticos.

# Tipos de requisitos Não Funcionais

**Figura 4.3**

Tipos de requisitos não funcionais



Devem descrever os requisitos funcionais e não funcionais de modo compreensível pelos usuários do sistema, que não tem conhecimentos técnicos detalhados.

Requisitos do usuário são definidos usando linguagem natural, tabelas e diagramas.

# Problemas com linguagem natural

---

## **Falta de clareza**

Difícil usar a linguagem de maneira precisa e sem ambiguidade, sem produzir um documento de difícil leitura

## **Confusão de requisitos**

Os requisitos funcionais e os não funcionais, os objetivos do sistema e as informações sobre o projeto podem não estar claramente definidos.

## **Fusão de requisitos**

Vários requisitos diferentes podem ser expressos juntos.

# Diretrizes para escrever requisitos

---

- Estabeleça um formato padrão e use-o para todos os requisitos.
- Utilize a linguagem de modo consistente. Faça distinção entre os requisitos obrigatórios e os que são desejáveis (O sistema **deve** ... O sistema **deveria** ...).
- Utilize destaque no texto para ressaltar partes importantes dos requisitos.
- Evite o uso de jargões de informática.



Em princípio, os requisitos deveriam definir o que o sistema deveria fazer e não como deveria ser implementado.

Na prática, é difícil conseguir essa separação

- Uma arquitetura inicial do sistema pode ser definida para ajudar a estruturar os requisitos.
- O sistema deve inter-operar com outros sistemas que geram requisitos para o novo sistema.
- O uso de um projeto específico pode ser um requisito externo de sistema

*User Stories* ou histórias de usuários tem o mesmo propósito dos casos de uso, mas não é a mesma coisa.

*User Stories* são descrições feitas pelos clientes, como funções, as quais gostariam que o sistema realizasse.

Elas também servem para estimar o plano de entrega das versões do sistema e para que sejam criados testes de aceitação de software. Estes testes indicam se o software cumpre mesmo o que cliente requisitou no seu plano inicial do sistema.

# **Documento de Requisitos**

# Documento de requisitos

---

- O documento de requisitos é a declaração oficial do que é exigido dos desenvolvedores de sistema.
- Deve incluir os requisitos de usuário e uma especificação detalhada dos requisitos de sistema.
- NÃO é um documento de projeto. Deve esclarecer o que o sistema deve fazer e não como deve ser feito.
- **O entregável do PI na disciplina Engenharia de Software**

# Requisitos de um Documento de requisitos

---

Deve:

- Especificar o comportamento externo do sistema.
- Especificar as restrições à implementação.
- Ser fácil de ser modificado.
- Servir como ferramenta de referência para a manutenção.
- Registrar a estratégia sobre o ciclo de vida do sistema.
- Caracterizar respostas aceitáveis para eventos indesejáveis.

# Padrão IEEE/ANSI 830 - 1993

---



- Introdução
- Descrição Geral
- Requisitos Específicos ( funcionais e não funcionais)
- Apêndices
- Índices

**Esta é uma estrutura genérica**

# Dúvidas

**Prof. Orlando Saraiva Júnior**  
**[orlando.nascimento@fatec.sp.gov.br](mailto:orlando.nascimento@fatec.sp.gov.br)**

Os **requisitos** estabelecem o que o sistema deve fazer e definem restrições sobre sua operação e implementação.

**Requisitos funcionais** são declarações de funções que o sistema deve fornecer.

**Requisitos não funcionais** se relacionam às propriedades emergentes do sistema e, portanto, se aplicam ao sistema como um todo.



**Requisitos de usuário** são declarações de alto nível do que o sistema deve fazer. Requisitos de usuário devem ser escritos em linguagem natural, tabelas e diagramas.

**Requisitos de sistema** se destinam a comunicar, de modo preciso, as funções que o sistema tem de fornecer.

O **documento de requisitos** de software é a declaração estabelecida dos requisitos do sistema.

# Prática