

# Requisitos de software

Uma visão introdutória

**Professor:**  
**Dr. Antônio Fernando Traina**

**[coordenacao.bigdata@fatecbb.edu.br](mailto:coordenacao.bigdata@fatecbb.edu.br)**

# Agenda

O que são Requisitos de software?

Requisitos funcionais e requisitos não funcionais

Especificação de Requisitos de Software

Técnicas de extração de requisitos

Comentários Finais

**O que são Requisitos de software?**

# Requisito

Requisito é um conceito que tem a sua raiz etimológica no latim *requīsitus*, palavra esta que, por sua vez, deriva do verbo latino “*requirere*”, que se pode traduzir como “reclamar” ou “requerer”.

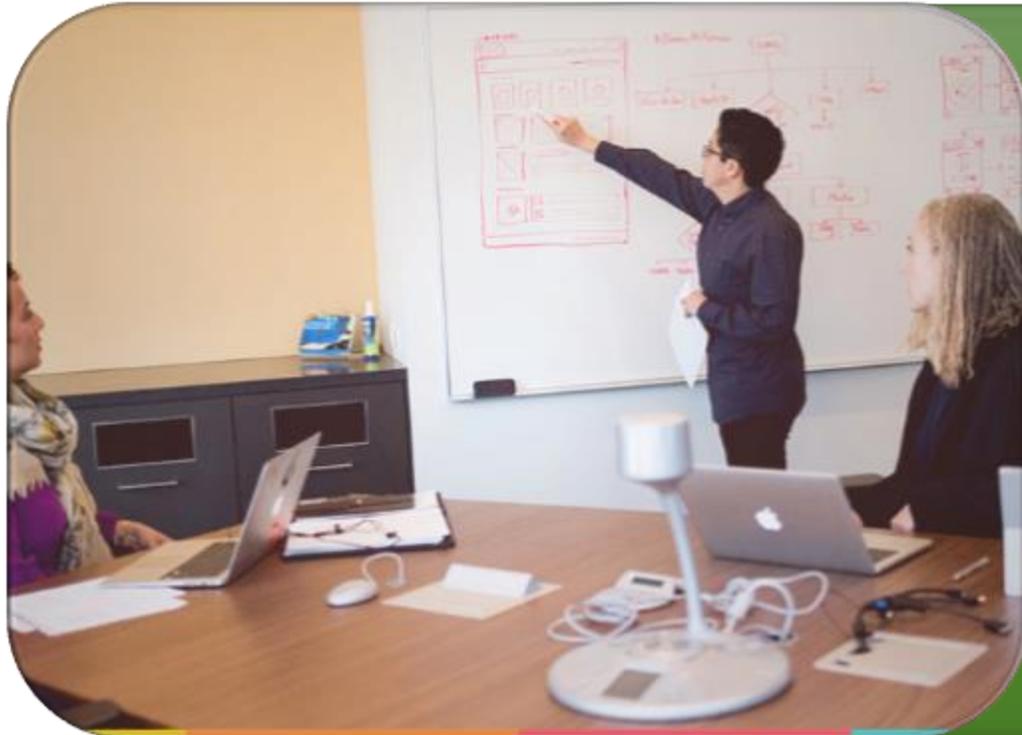
# Requisitos

Requisitos é um termo utilizado em todas as áreas, e que de uma forma geral descreve uma necessidade ou desejo, às vezes pessoal, às vezes de uma organização.



Tais desejos nem sempre são explícitos, documentados ou até mesmo claro para quem o deseja. É necessário muito diálogo e discussão sobre os Requisitos.

# Requisitos



Identificar e definir os requisitos, um papel que na área de informática é exercida pelo Analista de Requisitos, requer compreender as necessidades e o querer do negócio.

# Visões de requisitos



a) Requisitos do Negócio,



b) Requisitos do Usuário e



c) Requisitos de Software.

# Vamos descrever o conceito de cada um deles:

## 1-Requisitos do Negócio:

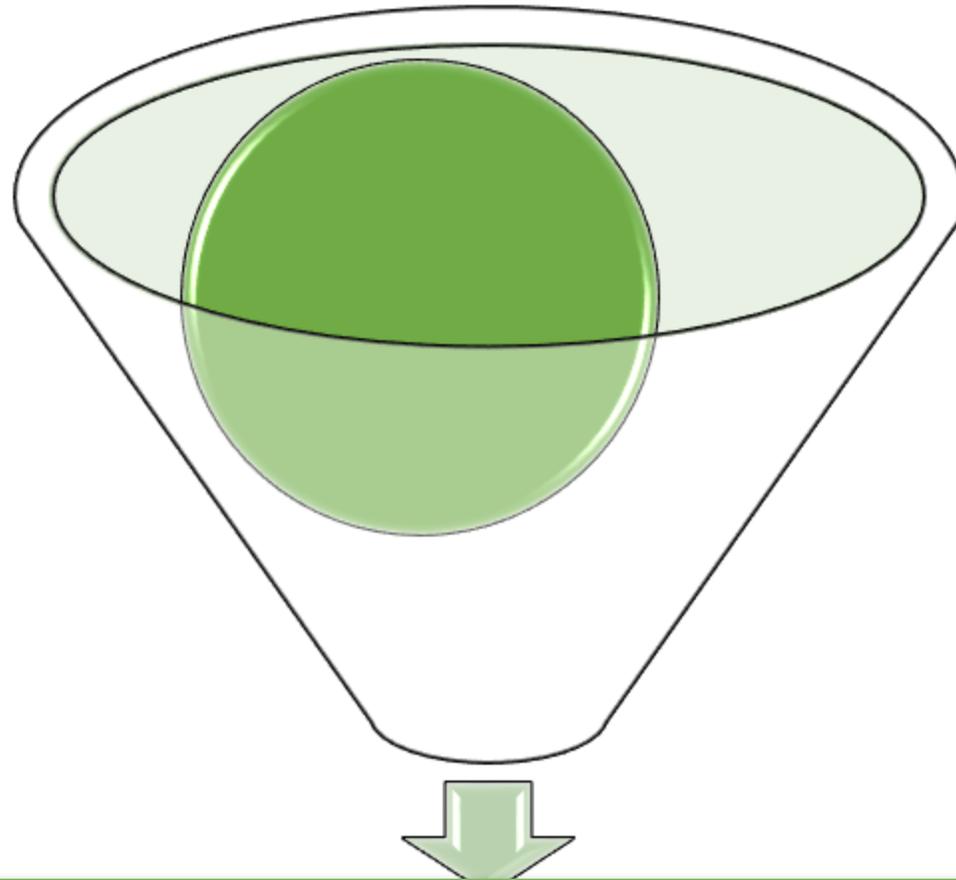
- Descrevem as necessidades do negocio que o software precisa atender, como por exemplo, prazo, custo, regras, alinhamento com os objetivos estratégicos, etc.

## 2-Requisitos do Usuário:

- Descrevem as necessidades do usuário do ponto de vista das tarefas a serem realizadas no software, definindo os objetivos geral e específicos, bem como as suas funcionalidades (visão macro dos requisitos).

## 3-Requisitos de Software:

- São as ações que o software deve executar, possuindo características e condições próprias, de forma a automatizar uma tarefa de um processo de negócio. Aqui definimos os requisitos funcionais e não funcionais, e, conforme o método IRON, Requisitos de Dados e Regras de Execução.



**As falhas em requisitos estão entre as principais razões para o fracasso de um software.**

# Requisitos de software

Entre as principais razões destacam-se:

- requisitos mal organizados,
- requisitos mal expressos,
- requisitos desnecessários para os clientes
- requisitos frequentemente mutáveis.

# Requisitos de software

Um **requisito** é definido como "uma condição ou um recurso com o qual um sistema deve estar em conformidade". Assim critérios devem ser estabelecidos para atender essas condições.

# Requisitos de software

Esses critérios podem ser de qualidade para o software, ou seja, os requisitos de performance, usabilidade, confiabilidade, robustez, etc.

Ou então, os critérios podem ser quanto a qualidade para o processo de software, ou seja, requisitos de entrega, implementação, etc.

# **Requisitos funcionais e requisitos não funcionais**

# Classificação dos Requisitos

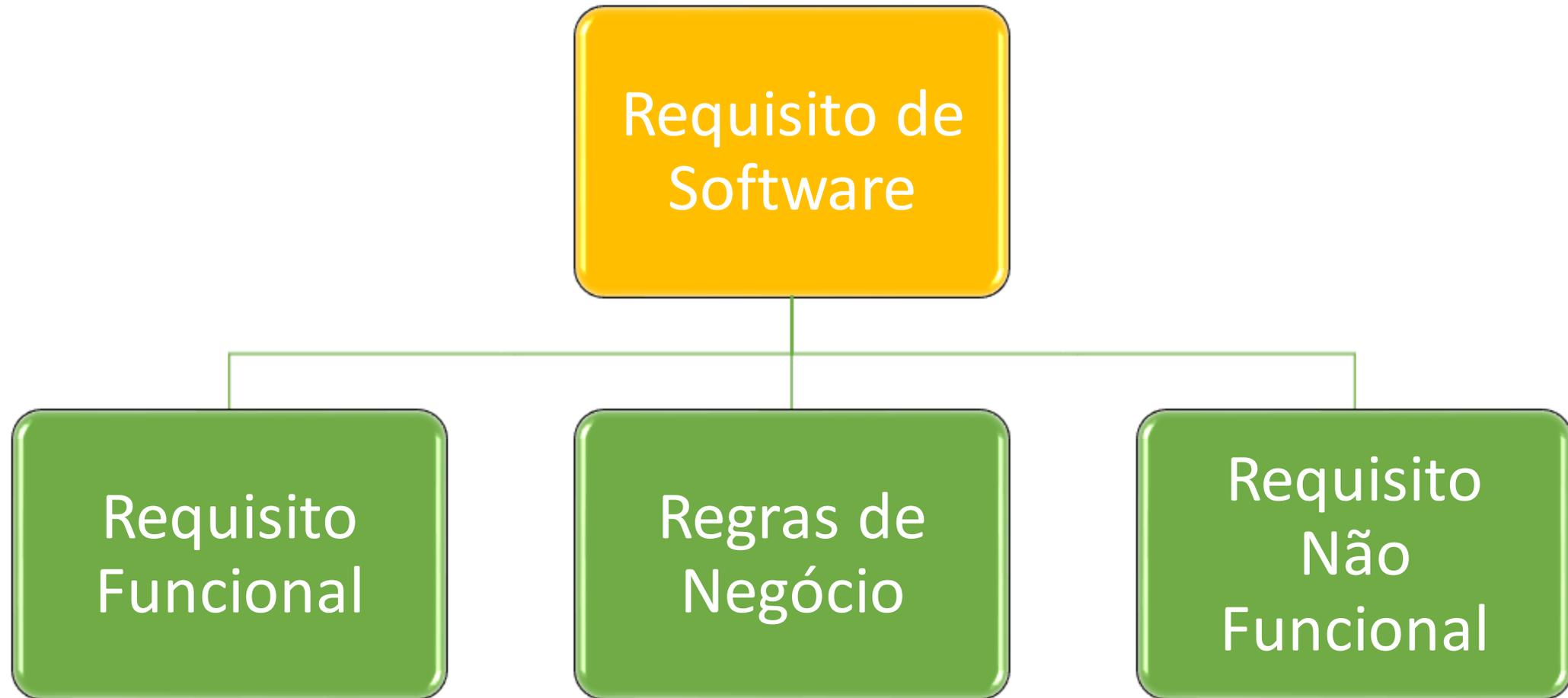
Existem dois tipos de classificação de requisitos, são eles:



Requisitos Funcionais (RF)

Requisitos Não-Funcionais (RNF).

# Requisito de Software

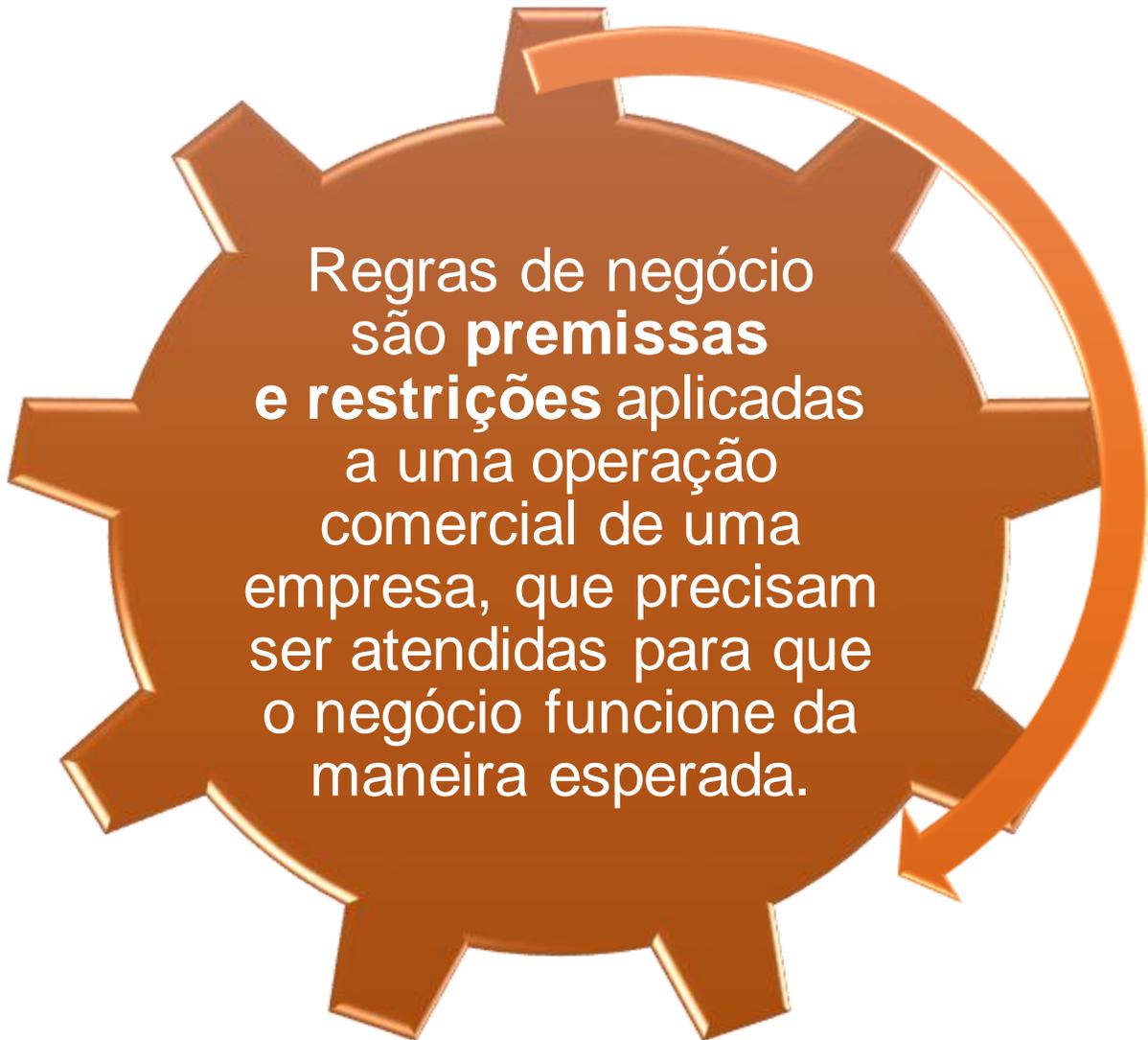


# Regra de negócio

É possível uma empresa mais arcaica **viver sem software**,  
mas não consegue **viver sem regras de negócio**.

Uma RN (Regra de Negócio), no contexto da Engenharia de Software, é tratada como um Requisito de Software, por ser algo que sem ela, o software não existe.

# Regra de negócio



Regras de negócio  
são **premissas**  
**e restrições** aplicadas  
a uma operação  
comercial de uma  
empresa, que precisam  
ser atendidas para que  
o negócio funcione da  
maneira esperada.

Mas os requisitos funcionais, como citado anteriormente, definem **quais são** as necessidades/exigências da empresa em termos funcionais (que funcionarão através de um sistema), ou seja, **o que o sistema deverá fazer**.

As regras de negócio definem como o sistema fará o atendimento às necessidades/exigências definidas; uma RN pode ser compreendida quanto a **como um requisito funcional se realizará**.

# Regra de Negócio: exemplo

Imaginemos uma empresa que possui um departamento de expedição de materiais.

Este departamento que **não possui software** para automatizar as atividades deste departamento.

Vejam os a seguir, um pouco sobre este cenário.

*Sempre que uma pessoa se dirigir ao departamento de expedição para solicitar uma mercadoria esta pessoa deve se identificar com seu documento de identidade. O profissional do departamento de expedição deve certificar-se que o documento é válido.*

*Após checar que o documento é válido, o profissional deverá pegar o documento de protocolo de entrega com a pessoa, e neste documento conterá a seção e caixa onde se encontra a mercadoria.*

*Deverá então dirigir-se à seção, na caixa identificada, pegar o material e levar ao guichê para entrega à pessoa que o solicitou. Antes de realizar a entrega, deverá solicitar que a pessoa assine o livro de entregas, incluindo seu documento e dados de endereço. No livro também devem ser escritos os dados da mercadoria (nome, categoria, marca e modelo), nome do profissional que fez a entrega, e data e hora da entrega.*

*Se a mercadoria solicitada não estiver na seção e caixa onde deveria estar, o profissional do departamento deverá entrar em contato com a gerência para reportar o problema. O mesmo deve ser feito caso identifique-se que o documento da pessoa que está buscando o material não é válido.*

# Regra de Negócio

Os critérios e restrições informados são regras, e regras da empresa (negócio) que faz as entregas.

Logo, são regras de negócio.

# Requisitos funcionais



Referem-se sobre o que o sistema deve fazer, ou seja, suas funções e informações.



Portanto, requisitos funcionais preocupam-se com a funcionalidade e os serviços do sistema, ou seja, as funções que o sistema deve fornecer para o cliente e como o sistema se comportará em determinadas situações.

# Requisitos funcionais

Seguem  
alguns  
exemplos de  
requisitos  
funcionais:

- [RF001] O Sistema deve cadastrar médicos profissionais (entrada)
- [RF002] O Sistema deve emitir um relatório de clientes (saída)
- [RF003] O Sistema deve passar um cliente da situação "em consulta" para "consultado" quando o cliente terminar de ser atendido (mudança de estado)
- [RF004] O cliente pode consultar seus dados no sistema

# Requisitos Não funcionais

Os requisitos não funcionais referem-se aos critérios que qualificam os requisitos funcionais.

Definem propriedades e restrições do sistema como tempo, espaço, linguagens de programação, versões do compilador, SGBD, Sistema Operacional, método de desenvolvimento, etc.

Uma dica importante é que os requisitos não funcionais são geralmente mensuráveis e assim devemos preferencialmente associar uma medida ou referência para cada requisito não funcional.

# Propriedades e métricas para requisitos não funcionais

Propriedade	Métrica
Velocidade	Transações processadas por segundo. Tempo de resposta ao usuário/evento. Tempo de atualização da tela.
Tamanho	Kbytes. Número de chips de RAM.
Facilidade de uso	Tempo de treinamento. Número de telas de ajuda.
Confiabilidade	Tempo médio para falhar. Probabilidade de indisponibilidade. Taxa de ocorrência de falhas. Disponibilidade.
Robustez	Tempo de reinício após falha. Porcentagem de eventos que causam falhas. Probabilidade de que os dados sejam corrompidos por falhas.
Portabilidade	Porcentagem de declarações dependentes de sistema-alvo. Número de sistemas-alvo.

# Atributos de um bom Requisito Funcional

Atributo	Referente a
<b>Unidade</b>	O RF deve propor uma 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.
<b>Compleitude</b>	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”.
<b>Consistência</b>	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.

# Atributos de um bom Requisito Funcional

Atributo	Referente a
<b>Atomicidade</b>	<p>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”.</p>
<b>Não-Ambiguidade</b>	<p>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”.</p>

# Atributos de um bom Requisito Funcional

Atributo	Referente a
<b>Verificável</b>	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.
<b>Rastreável</b>	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.).
<b>Prioridade</b>	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.

# Matriz de Rastreabilidade

Requisitos X Casos de Uso	[UC-001] - Incluir Recebimento	[UC-002] - Alterar Recebimento	[UC-003] – Validar Contas Recebidas	[UC-004] – Conciliar Contas	[UC-005] - ...
[RF-001] – Recebimento de Conta	X	X			
[RF-001] - Transmissão de Contas Recebidas			X		
[RF-003] - Transmissão de Depósitos				X	
[RF-010] – Conciliação Fiscal Automática					
[RF-011] - ...					

# Matriz de Rastreabilidade

www . Fabio Cruz . com

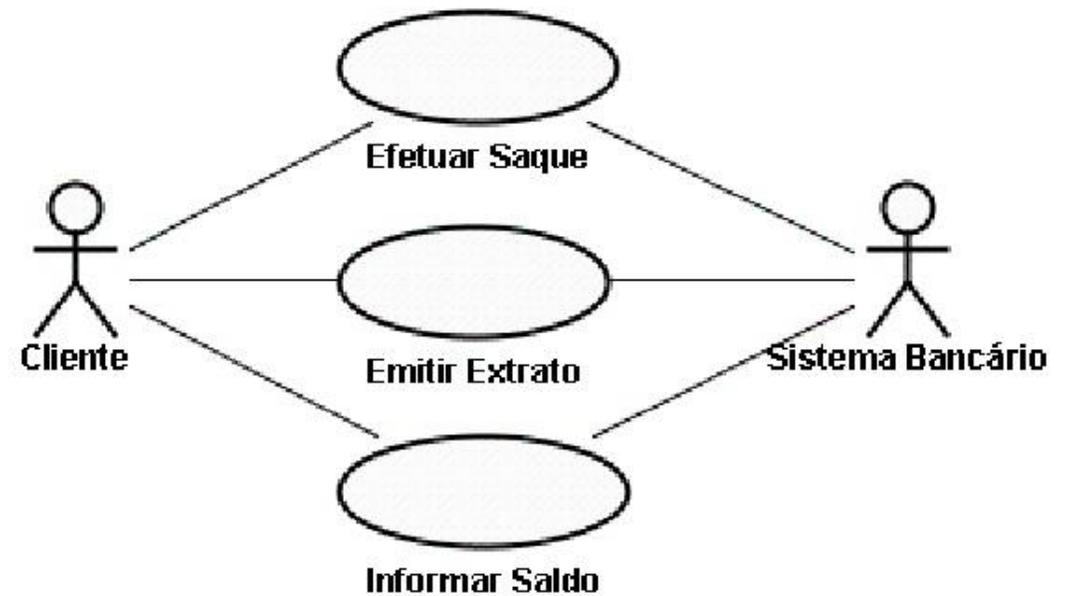
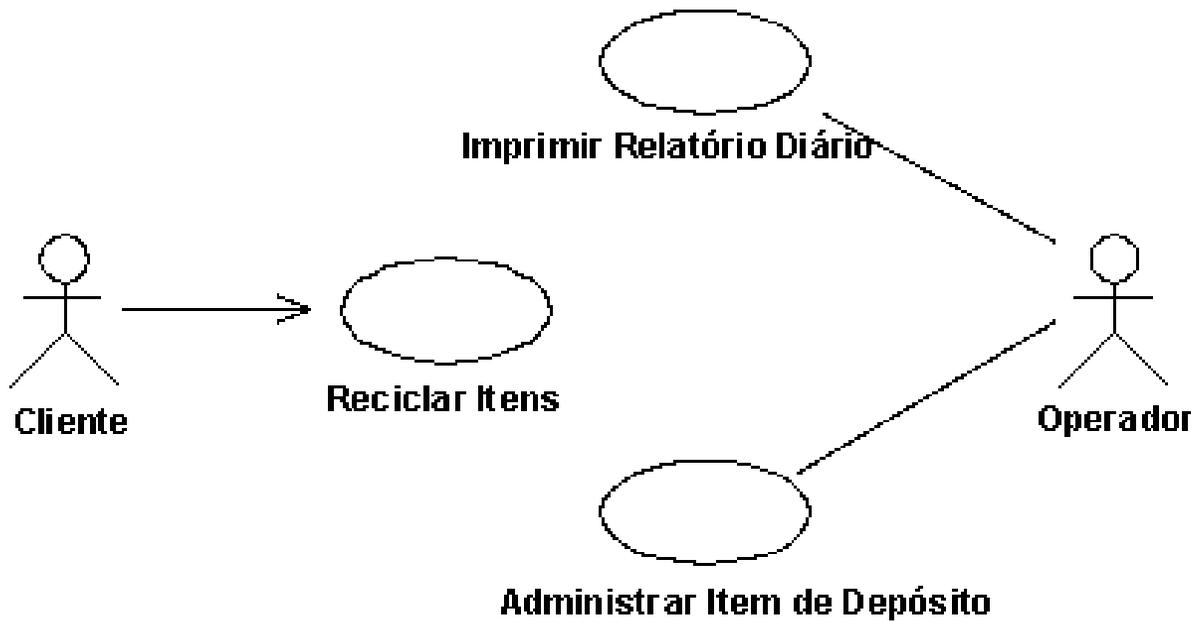
**Matriz de rastreabilidade de requisitos - versão 1.1**

	Funcionalidade 1	Funcionalidade 2	Funcionalidade 3	Funcionalidade 4		
Requisito 1	x					
Requisito 2		x	x			
Requisito 3				x		
Requisito 4				x		

**Projeto A**

Requisitos Funcionais	Casos de Uso			
	UC-1	UC-2	UC-3	UC-4
FR-1	↙			
FR-2	↙			
FR-3			↙	
FR-4			↙	
FR-5		↙		↙
FR-6			↙	

# Exemplos de Caso de Uso (Use case)



# Estrutura de um Requisito Funcional

<b>Identificador</b>			
<b>Nome</b>			
<b>Módulo</b>			
<b>Data de criação</b>		<b>Autor</b>	
<b>Data da última alteração</b>		<b>Autor</b>	
<b>Versão</b>		<b>Prioridade</b>	
<b>Descrição</b>			

# Onde cada campo:

<b>Campo</b>	<b>Descrição</b>
<b>Identificador</b>	Sufixo seguido de um identificador único. O sufixo geralmente utilizado é RF (Requisito Funcional) e o identificador único geralmente é composto de quatro dígitos.
<b>Nome</b>	Nome curto do RF, mas que possibilite entender bem o que RF faz apenas pelo nome.
<b>Módulo</b>	Módulo ao qual o RF pertence. Se for um sistema pequeno que não possua nenhum módulo, somente o próprio sistema, deve ser preenchido com N/A (não se aplica).
<b>Data de criação</b>	Data da criação do RF, ou a data em que ele foi especificado.

# Onde cada campo:

<b>Campo</b>	<b>Descrição</b>
<b>Autor</b>	Profissional que especificou o RF pela primeira vez, quem o criou.
<b>Data da última alteração</b>	Data em que houve a última alteração no RF.
<b>Autor</b>	Profissional que alterou a especificação do RF pela última vez.
<b>Versão</b>	Número da versão do RF. Geralmente utiliza-se algo simples, como 1, 2. A versão inicial sempre é a 1, e a cada alteração incrementa-se a versão (na criação versão 1, na primeira alteração versão 2 e por aí vai).
<b>Prioridade</b>	Se o RF é Essencial, Importante ou Desejável.
<b>Descrição</b>	Descrição detalhada (a mais detalhada possível) do RF.

# Exemplo 1

<b>Identificador</b>	RF0003		
<b>Nome</b>	Emitir carta de cobrança para clientes inadimplentes conforme critérios pré-estabelecidos		
<b>Módulo</b>	Cobrança		
<b>Data de criação</b>	31/01/2016	<b>Autor</b>	Hípias de Elis
<b>Data da última alteração</b>	N/A	<b>Autor</b>	N/A
<b>Versão</b>	1	<b>Prioridade</b>	Essencial
<b>Descrição</b>	<p>Para todo cliente que esteja inadimplente deverá ser possível a emissão de carta de cobrança através do sistema. O resultado da emissão será a carta impressa, que será posteriormente despachada por correios ou outro agente de entrega.</p> <p>Os critérios para definir se um cliente é ou não inadimplente devem estar cadastrados no sistema utilizando regras de negócio específicas. Nem todo cliente com pagamento em atraso será considerado inadimplente, pois deverá ser levado em consideração o score de crédito do cliente, sua renda mensal, patrimônio etc. Obs.: verificar os Requisitos Funcionais e regras de negócio específicas para manutenção dos critérios citados.</p> <p>Não haverá diferença na emissão da carta de cobrança para clientes pessoa física (particular) ou jurídica (empresarial), logo, o processo será o mesmo para ambos.</p>		

**Exceção** O registro da avaliação não poderá ser concluído caso o usuário deixe de preencher algum campo.

## **RS003 Registro de avaliação de turma**

**Referência** [Registrar avaliação de turma por docente.RF003], [Impedir acesso direto ao SIGA.RF006], [Bloquear acesso direto ao SIGA.RF005]

**Sumário** O caso de uso é responsável por registrar a avaliação de turma.

**Pré-condições** O usuário deve estar ministrando as turmas das disciplinas avaliadas no período, o período de avaliação de turmas deve estar aberto.

**Atores** Docente

- Descrição**
1. O usuário faz login no SIGA.
  2. O sistema exibe uma tela, antes da tela principal do SIGA, contendo um formulário em forma de matriz de perguntas (linhas) x código de turma (colunas), um botão para “registrar avaliação” e um botão para “responder depois”. As turmas exibidas são todas as ministradas pelo docente no período avaliado.
  3. O usuário faz clique sobre o botão responder.
  4. O sistema registra avaliação.

**Alternativas** No passo 2, o botão “responder depois” ficará desativado a partir de determinada data de acordo com XYZ

**Exceção** O registro da avaliação não poderá ser concluído caso o usuário deixe de preencher algum campo.

# Requisitos Não funcionais

Os requisitos não funcionais ainda são classificados em três tipos, são eles:

- Requisitos do Produto Final,
- Requisitos Organizacionais
- Requisitos Externos.

Requisitos do Produto Final

- Referem-se a como o produto deve comportar-se, ou seja, a sua velocidade de execução, confiabilidade, etc.

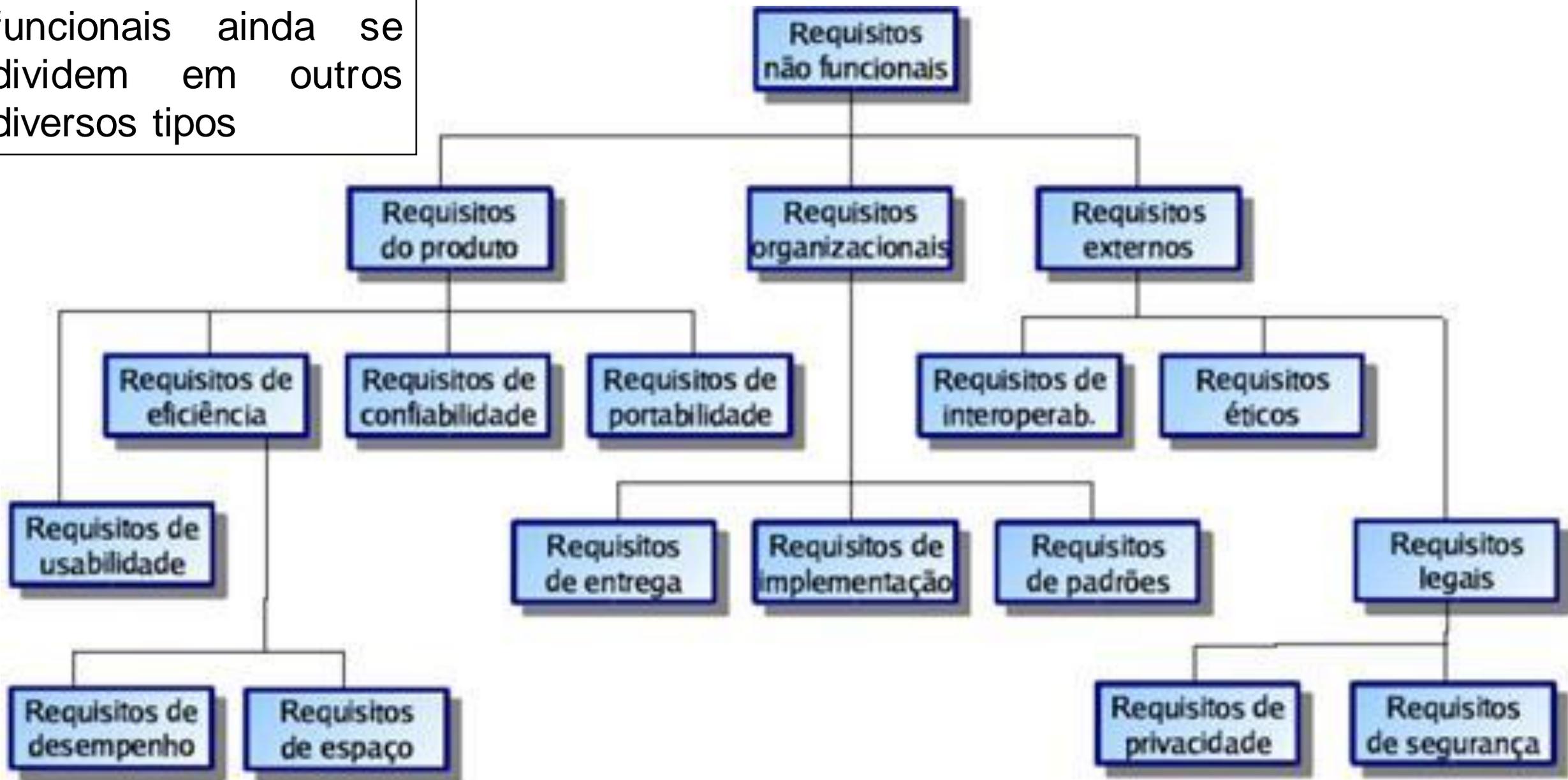
Requisitos Organizacionais

- Referem-se à consequência de políticas e procedimentos organizacionais que devem ser seguidos.

Requisitos Externos

- Referem-se a fatores externos ao sistema e ao processo de desenvolvimento como a legislação.

Os requisitos não funcionais ainda se dividem em outros diversos tipos



# Requisitos Não funcionais

Seguem  
alguns  
exemplos de  
requisitos  
não  
funcionais:

- [RNF001] O sistema deve imprimir o relatório em até 5 segundos.
- [RNF002] Todos os relatórios devem seguir o padrão de relatórios especificado pelo setor XYZ.
- [RNF003] O sistema deve ser implementado em Java.

# Requisitos de software

Existem vários tipos de requisitos.

Uma maneira de categorizá-los é descrita como o modelo **FURPS+**, conforme mostrado a seguir.

- **Funcionalidade**
- **Utilidade**
- **Confiabilidade**
- **Desempenho**
- **Suportabilidade**

# Requisitos de software

O + sugere  
se deve-se  
incluir  
requisitos  
como:

- restrições de Design
- requisitos de implementação
- requisitos de interface
- requisitos físicos

# Funcionalidade

Os  
requisitos  
funcionais  
podem  
incluir:

- conjuntos de recursos
- recursos
- segurança

# Utilidade

Os requisitos de utilidade podem incluir subcategorias como:

- fatores humanos
- estética
- consistência na interface com o usuário
- ajuda on-line e sensível ao contexto
- assistentes e agentes
- documentação do usuário
- materiais de treinamento

# Confiabilidade

Os requisitos de confiabilidade a serem considerados são:

- frequência e gravidade de falha
- possibilidade de recuperação
- possibilidade de previsão
- precisão
- tempo médio entre falhas (MTBF)

# Desempenho

Um requisito de desempenho impõe condições aos requisitos funcionais. Por exemplo, para uma determinada ação, ele pode especificar parâmetros de desempenho para o seguinte:

- velocidade
- eficiência
- disponibilidade
- precisão
- produtividade
- tempo de resposta
- tempo de recuperação
- uso de recurso

# Suportabilidade

Os requisitos  
de  
suportabilidade  
podem incluir:

- possibilidade de teste
- extensibilidade
- possibilidade de adaptação
- possibilidade de manutenção
- compatibilidade
- possibilidade de configuração
- possibilidade de serviço
- possibilidade de instalação
- possibilidade de localização (internacionalização)

# Requisito de Implementação

Um requisito de implementação específica ou restringe o código ou a construção de um sistema. Como exemplos, podemos citar:

- padrões obrigatórios
- linguagens de implementação
- políticas para integridade de banco de dados
- limites de recursos
- ambientes de operação

# Requisito de Design

Requisito  
de design

- Freqüentemente chamado de **restrição de design**, especifica ou restringe o design de um sistema.

# Requisito de Interface

Um  
requisito  
de  
interface  
especifica:

- um item externo com o qual o sistema deve interagir
- restrições de formatos, tempos ou outros fatores utilizados por tal interação

# Requisito Físico

Um requisito físico especifica uma característica física que um sistema deve possuir, por exemplo:

- material
- forma
- tamanho
- peso

Esse tipo de requisito pode ser utilizado para representar requisitos de hardware, como as configurações físicas de rede obrigatórias.

# **Especificação de Requisitos de Software**

# Especificação de Requisitos de Software

Muitos pensam que os requisitos só devem ser especificados no início do projeto, o que não é verdade, pois os requisitos devem ser considerados durante todo o ciclo de vida de desenvolvimento do software.

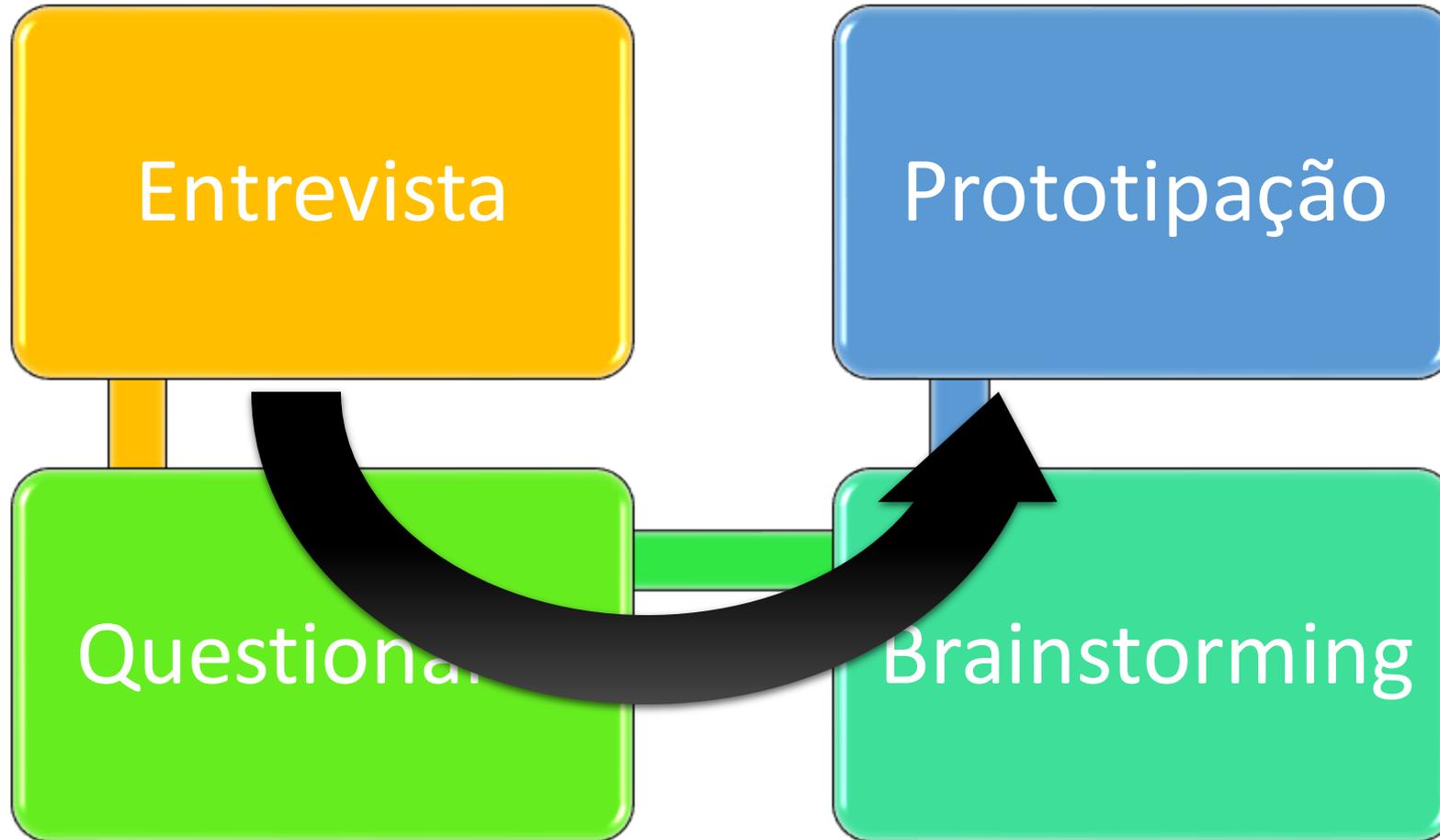
# Especificação de Requisitos de Software

Segue abaixo as etapas onde os requisitos são utilizados durante o ciclo de vida do projeto:

- Definição de critérios de aceitação e validação (pelos stakeholders);
- Definição sobre O QUÊ o sistema deve fazer (especificação para a equipe);
- Teste e Verificação do sistema sendo desenvolvido;
- Informações para gerenciamento de mudanças (Rastreabilidade, análise de impacto);
- Alocação de tarefas para a equipe;
- Estimativa de custo/esforço/cronograma;
- Acompanhamento e Controle do andamento do projeto.

# Técnicas de extração de requisitos

# Técnicas



# Entrevista



Diálogo entre entrevistador e entrevistado.



Trata-se uma situação social objetiva, pois permite que os envolvidos interajam de maneira mais completa.

# Entrevista

Os participantes compartilham a situação, porém o entrevistador é responsável pela consecução do objetivo, que pode ser:

- Examinar a emotividade da pessoa entrevistada, seus anseios, tendências, atitudes e motivação;
- Ouvir um relato, biográfico ou cronológico, de fatos vivenciados ou observados pelo entrevistador;
- Utilizar a entrevista para coordenar informações recebidas de fontes diversas, podendo apreciar o entrevistado e seu conjunto.
- Para a preparação da entrevista o responsável deverá:

# Entrevista - Para a preparação da entrevista o responsável deverá:

Fixar os objetivos da entrevista, os quais devem ser claramente definidos, pois determinam o que se pretende para satisfazer as necessidades identificadas de informação;

Procurar conhecer o entrevistado, características, experiências anteriores que podem influenciar na entrevista;

Determinar perguntas objetivas, não permitindo múltiplas interpretações;

Estudar o assunto, recorrendo a livros e/ou manuais;

Definir a forma de abordagem.

# Questionário



A utilização de um questionário se aplica quando:

- A distância é considerável;
- Se queira obter um primeiro conhecimento

# Questionário: os benefícios de se utilizar o questionário são:

Economia de tempo;

Baixo custo;

Anonimato;

Tabulação;

Pesquisa

# Questionário

A preocupação na preparação de um questionário é quanto à elaboração do mesmo, que deve ser feita de forma clara com perguntas encadeadas e complementares.

# Brainstorming

Buscar soluções através de geração espontânea de ideias, onde as regras são:

- É permitido ter ideias;
- Não é permitido julgar ou criticar ideias
- Ambiente é não avaliativo.

# Brainstorming



Tem como base a descontração, o pensamento de que quanto mais ideias melhor, o enriquecimento da ideia alheia e o evitamento de discussão de ideias.



Deve ser definido o papel do condutor e do escriba, este anota cada ideia em quadro, flip-chart, ou outro.

# Brainstorming: um brainstorming deve ter como métodos:

Divulgação clara dos objetivos;

Geração de ideias (entre 20 a 60 minutos);

Intervalo para relaxamento

Estudo detalhado de cada ideia

# Prototipação

Utilizado no estágio inicial do projeto.

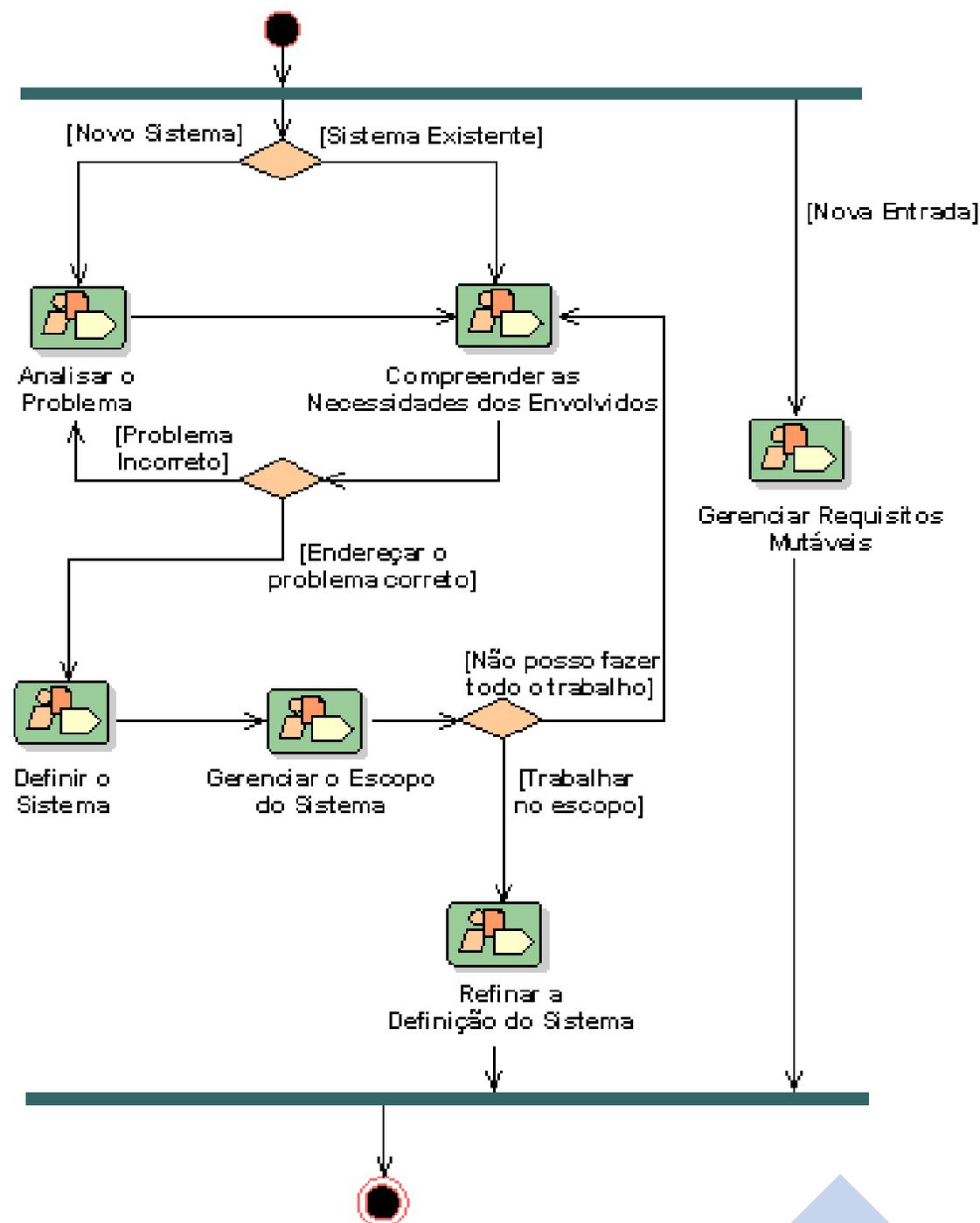
Ajuda aos stakeholders a desenvolver uma forte noção sobre a aplicação a qual ainda não foi implementada, que através da visualização da mesma eles podem identificar os reais requisitos e fluxos de trabalho do sistema.

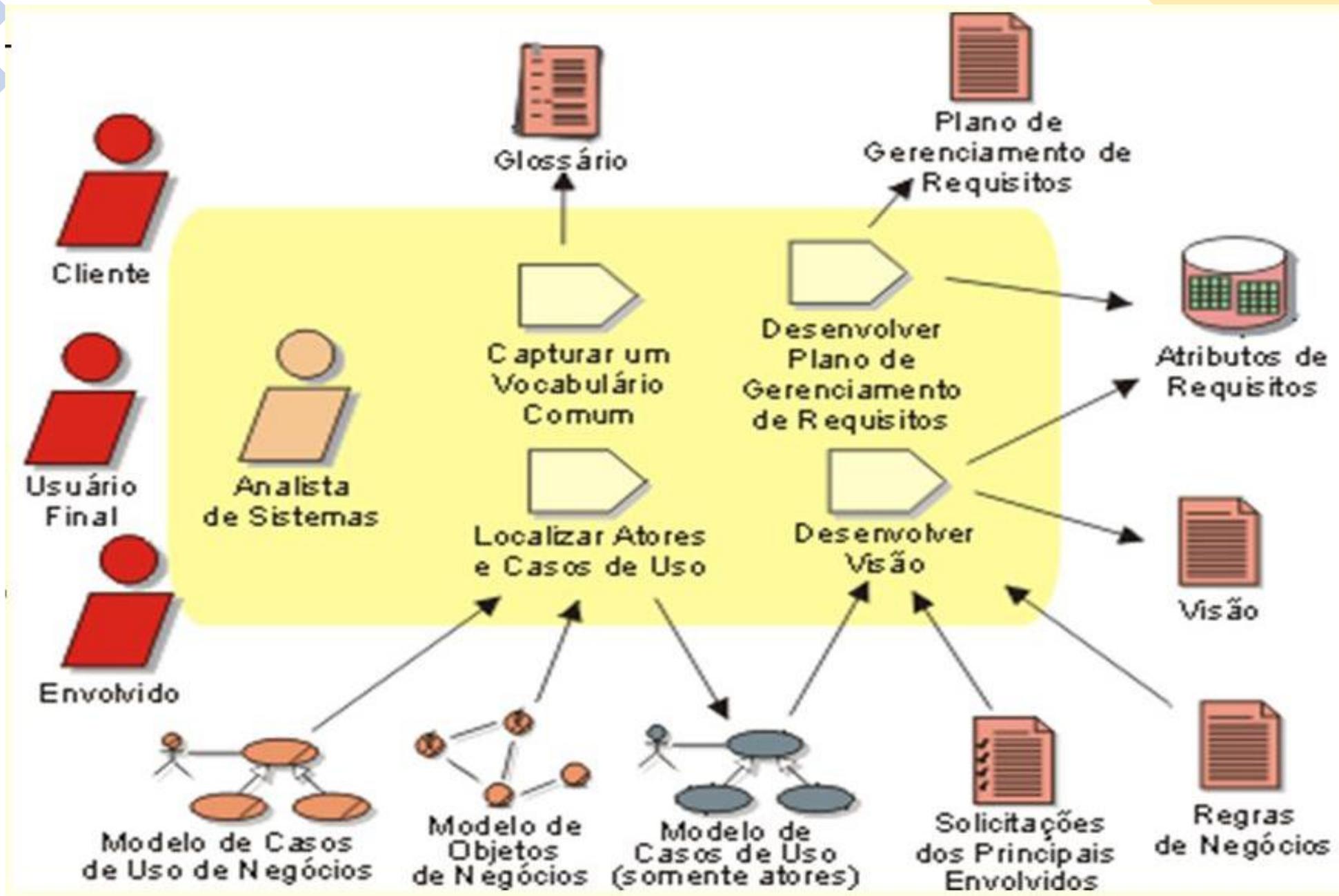
É muito utilizado quando os stakeholders são incapazes de expressar os seus requisitos ou se os mesmos não têm nenhuma experiência com o sistema.

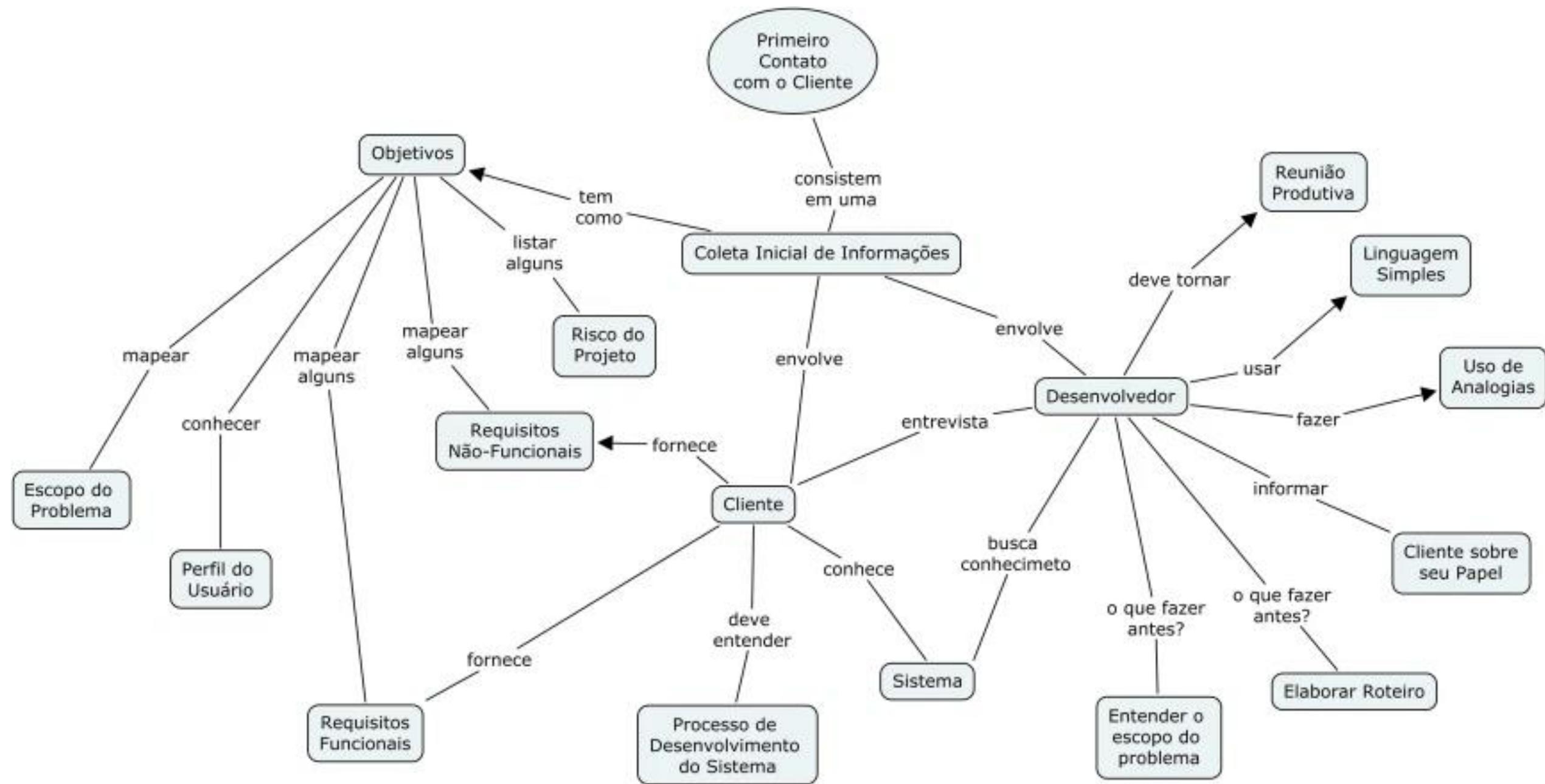
# Comentários Finais

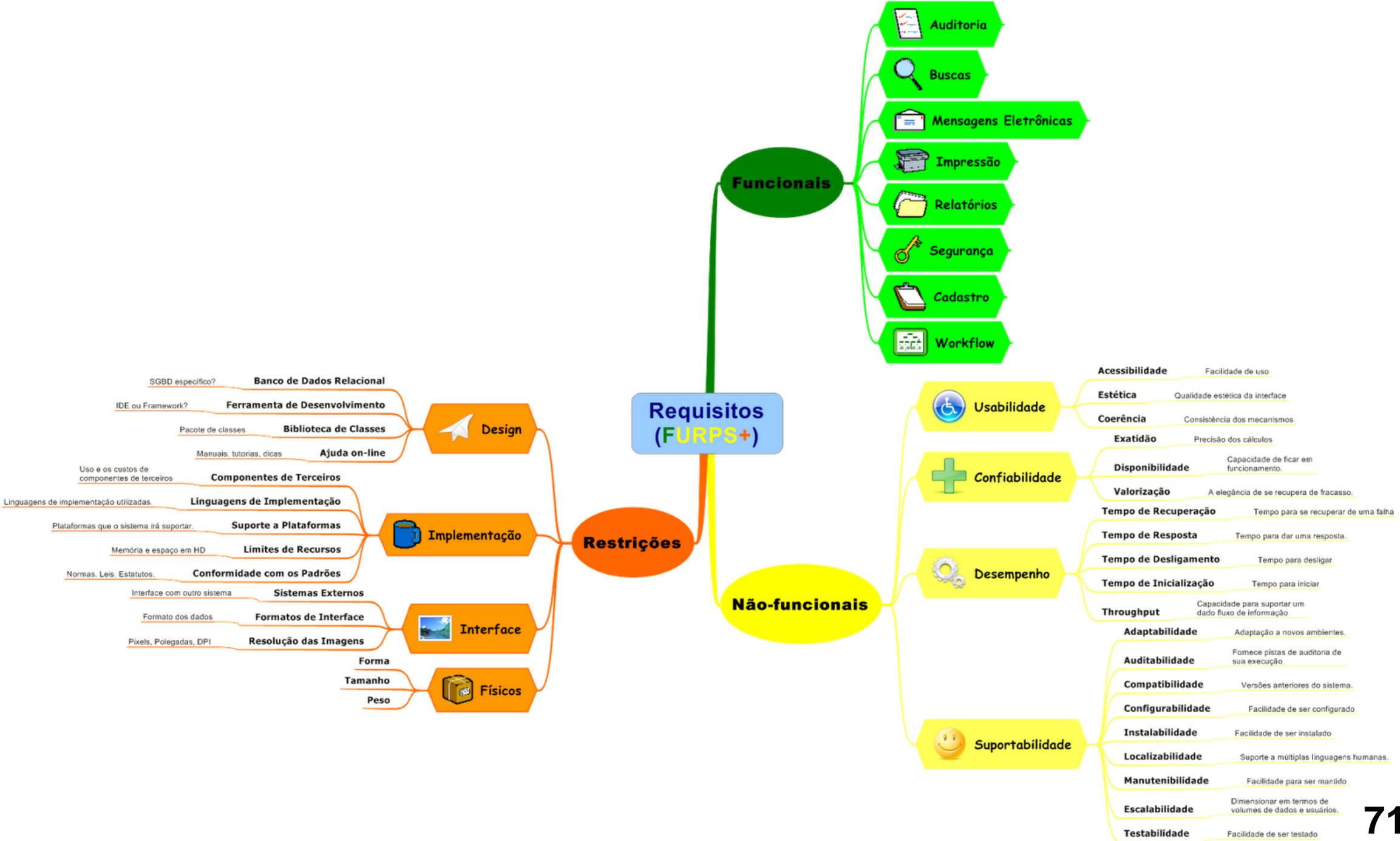
# Roteiro simplificado



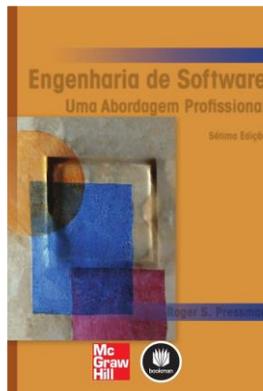




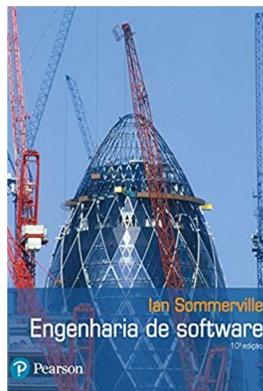




# Bibliografia Básica



PRESSMAN, R. S.  
*Engenharia de Software.*  
7.ed. McGraw-Hill, 2006.



SOMMERVILLE, I.  
*Engenharia de Software.*  
8.ed. Addison Wesley, 2007.

# Referência eletrônica



<https://www.indtech.com.br/>



[tp://mds.cultura.gov.br/core.baseup/guidances.html](http://mds.cultura.gov.br/core.baseup/guidances.html)

**Fim**