

# **REQUISITOS DE SISTEMAS**

## **IDENTIFICAÇÃO DE STAKEHOLDERS**

# Olá!

Nesta aula, você irá:

- 1 - Conhecer o conceito de stakeholders.
- 2 - Identificar características dos stakeholders.
- 3 - Conhecer técnicas de levantamento de requisitos.
- 4 - Relacionar cenários distintos e as melhores técnicas de levantamento de requisitos a serem aplicadas.

## 1 Introdução

Uma vez que já definimos uma estrutura e classificação para o desenvolvimento de um projeto de software com controle e garantia da qualidade, vamos na aula de tratar de identificar e detalhar sobre as pessoas que tenham e /ou sejam afetados por um software. Porque, também independente de tecnologia, certamente um projeto que atua na área da engenharia de software, contemplará a participação direta ou indireta, ativa ou passiva, de pessoas.

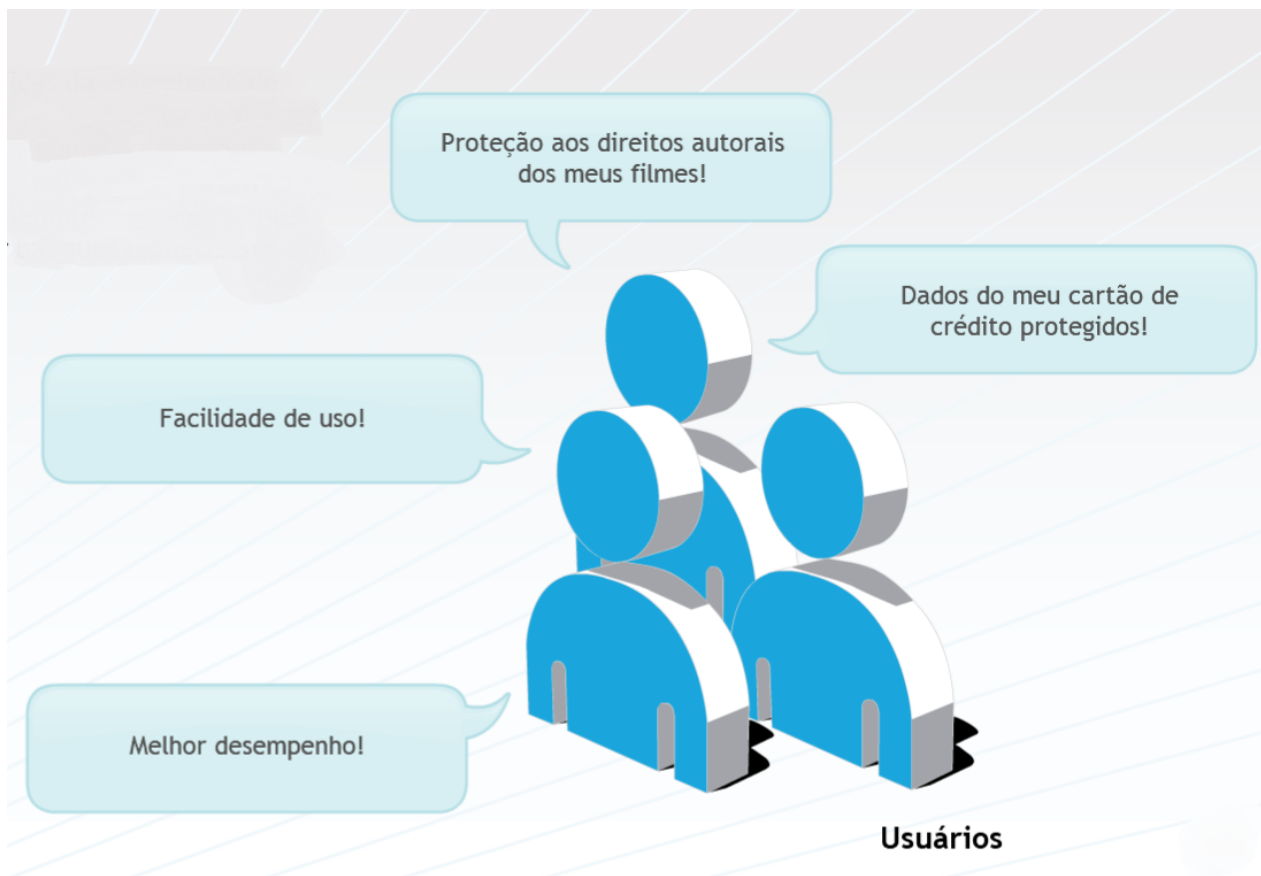
E justamente nesse perfil de um indivíduo que em algum momento influi algum tipo de interesse, participação, etc., sobre um determinado software, a este caracterizamos como um stakeholder.

## 2 Stakeholder

Antes de levantarmos as considerações específicas da engenharia de software, vamos buscar um entendimento mais amplo do que vem a ser um stakeholder, permitindo assim que possamos compreender, por exemplo, que não é somente na Computação que existem, mas em todo em qualquer sistema (computacional ou não), sempre teremos agentes diretos ou indiretos, que irão compor ou sofrer das suas características.

**Atenção:** Acompanhando ao abaixo, tente identificar a atividade da empresa que possa conter todos os usuários com base nos “pensamentos” citados nas caixas. Também procure distinguir quais deles seriam colaboradores e quais representam o(s) cliente(s), e, principalmente, sinalize quais serão os stakeholders desse cenário.

Deixe anotada a sua resposta para verificar se está correta ou completa.



Com foco então em um ciclo de vida do software é possível claramente saber que ele é composto por diversas e distintas responsabilidades que estão vinculadas as pessoas, grupos e entidades. Dentre essas responsabilidades que podemos aqui elucidar, são exemplos: o responsável pelo financiamento, o projeto, o desenvolvimento, o teste, o uso e a manutenção do software. A todos estes chamamos de **stakeholder**.

A palavra vem de da seguinte composição:

**Stake:** interesse, participação, risco.

**Holder:** aquele que possui.

**Atenção:** Portanto, todos aqueles que de alguma maneira é afetado pelo software, é um stakeholder (reveja sua resposta da questão vinculada ao questionamento da tela anterior). Então é correto afirmar que toda a preocupação no tocante ao correto desenvolvimento de um sistema e demais recursos de infraestrutura, por sua vez, tem o duplo objetivo de facilitar o cumprimento das responsabilidades dos stakeholders, quanto atender às suas **necessidades** (Voltando a ilustração temos: a urgência por desempenho, os aspectos de segurança e usabilidade).

**Segundo Sommerville (2009), podemos definir da seguinte forma:**

“Um stakeholder em uma arquitetura de software é uma pessoa, grupo ou entidade com um interesse ou preocupações sobre a realização da arquitetura.”

## 2.1 Exemplos de Stakeholders

Com base então no que já aprendemos, podemos então relacionar os seguintes exemplos e atribuições de stakeholders no desenvolvimento de um projeto de software:

- **Gerente de projeto**

Responsável em organizar e conduzir as equipes em suas responsabilidades. Como gestor, precisa manter harmonia no desenvolvimento do projeto, supervisionando a execução das tarefas, observar os processos, sustentar e fomentar o equilíbrio entre os stakeholders, etc.

- **Analista de Sistema**

Responsável em analisar quais as características o que deverá ter o produto a ser desenvolvido para atingir o objetivo final, ou seja, o que o cliente espera. Para isso, busca analisar as especificidades inerentes ao determinado software.

- **Programador**

São os responsáveis em efetivamente desenvolver o software. Cabe a ele a implementação das linhas de códigos que construirão a identidade lógica do software.

- **Patrocinador**

Popularmente é quem “paga a conta”. É aquele que libera os recursos, custeia a produção do projeto. Ele será o responsável por prover financeiramente a arquitetura necessária para o desenvolvimento de software.

- **Cliente (usuário)**

É aquele que, a partir de uma necessidade, faz a encomenda de um software. Portanto, é quem vai usufruir do produto a ser entregue. seja ele apenas um ou um grupo de usuários.

Contudo, além destes supracitados, podemos ter muitos outros stakeholders – que não são tão elementares, mas possuem algum tipo de interesse. Por exemplo:



Para contextualizar melhor, acompanhe o exemplo abaixo:

No contexto do início dessa aula, podemos dizer que uma pessoa qualquer que ficou no carro aguardando o amigo ou parente devolver um filme, tem grande interesse no desempenho do sistema utilizado pela locadora. caso contrário, o tempo de espera será maior do que o previsto. Portanto, tanto quem está no carro aguardando, como quem está na locadora para devolver o filme, são stakeholders.

## 2.2 Características dos Stakeholders

São outras características dos stakeholders:

- São específicos para cada projeto.
- Possuem anseios e objetivos distintos em um projeto.
- São atores fundamentais para detalhamento do que deve ser desenvolvido.

Mediante então a todos esse envolvimento, fato que pode atrapalhar consideravelmente o êxito de um projeto de software é a ocorrência de uma falha no levantamento dos stakeholders. Tal realidade pode representar que o gerente de projeto não estará pensando nas necessidades de todos os envolvidos. como consequência, podemos ter um software que não atende aquilo que o cliente esperava, e de que deverá ser revisto e ajustado posteriormente. Portanto, gerará desgastes de recursos, além da insatisfação daquele que encomendou o sistema. Entretanto, o gerente de projeto deve observar bem seus objetivos e não procurar stakeholders por todos lados, o que culminará em um cenário difícil de gerenciar. Deve haver limitação no escopo daqueles que afetam e/ou serão afetados pelo projeto. Caso contrário, com um pouco de imaginação, pode-se considerar stakeholder até a mulher do gerente de projeto que deixará de sair com o marido no fim de semana porque ele terá que trabalhar!

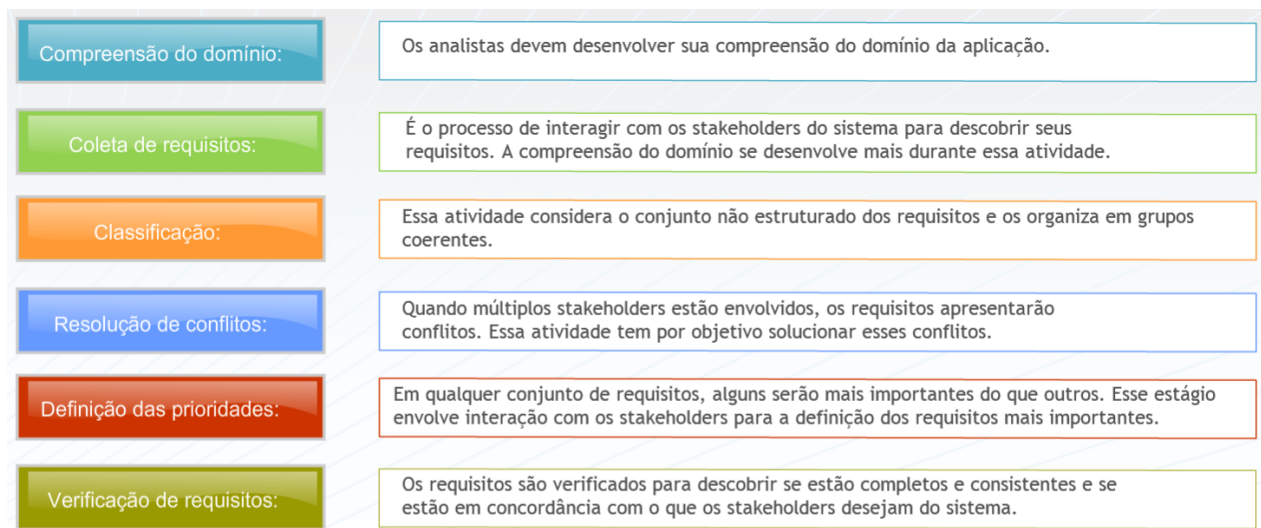
**Atenção:** Concluindo essa primeira etapa de nossa aula, esperamos que você possa ter entendido toda o envolvimento e influência dos stakeholders em um projeto de software, suas relações e inter-dependência na concepção e uso de um determinado sistema.

## 3 Técnicas para levantamento de Requisitos

Diferentemente do que pode representar a melhor das iniciativas, acionar os programadores para começarem a escrever linhas de códigos não deve ser o foco de atenção para o projeto de software, a abertura para toda a atividade de desenvolvimento de software deve ser o levantamento de requisitos.

Como já sabemos, a fonte das informações inerentes ao que queremos automatizar está com os futuros usuários, os quais atualmente já desenvolvem as mesmas tarefas, de forma manual ou automática. Além que o cliente não estabelece claramente todas as regras relativas ao negócio. e quem está sendo contratado desconhece as especificidades referente ao processo que atualmente está em execução.

Sommerville (2009) propõe um processo genérico de levantamento e análise que contém as atividades que veremos a seguir.



Apesar de aparentemente parecer elementar, o problema de não saber especificar corretamente o que o sistema deverá fazer é muito rotineiro. Realidades como:

- (a) de um usuário principal do sistema não saber o que quer que o sistema faça ou sabe e não consegue/quer transmitir para o analista. ou
- (b) requisitos identificados, mas que não exprimem a realidade. e
- (c) não estão em concordância com os requisitos informados por pessoas diferentes, são constantes na elaboração dos requisitos.

**Atenção:** Um stakeholder ou informação errada afetará em perda de tempo e dinheiro para ambas as partes envolvidas no desenvolvimento do sistema. É preciso aferir e revisar os requisitos.

No tocante ao do analista responsável pelo levantamento dos requisitos, dois fatores contribuem para o baixo grau de satisfação dos usuários, o que aumenta consideravelmente a perspectiva do erro. São eles:

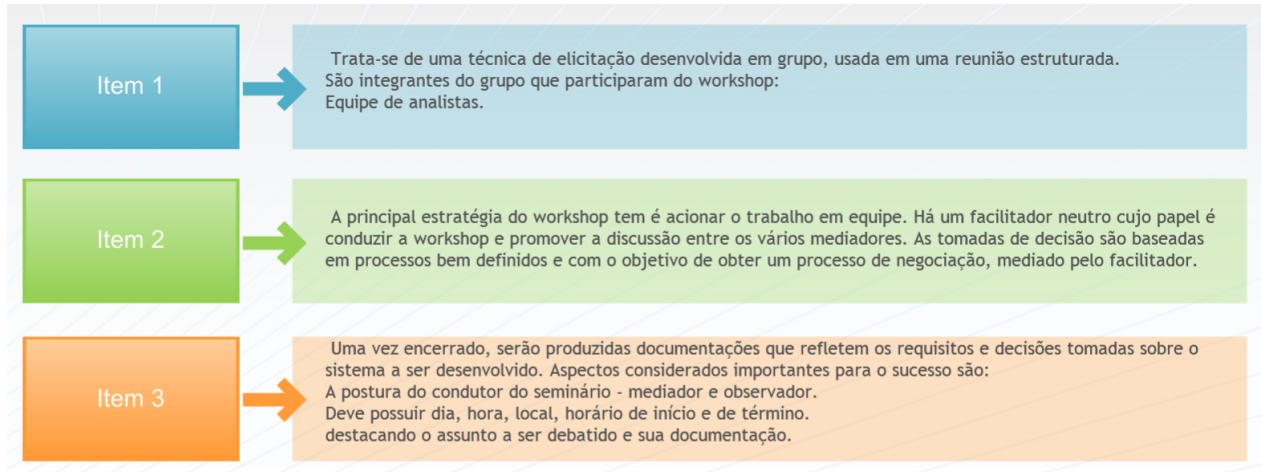
- Não utiliza uma técnica adequada para extrair os requisitos do sistema.
- Descrever os requisitos do sistema de modo pouco claro, com ambigüidades, sem consistência com todos os aspectos significativos do sistema proposto.

Como contramedida para atacar primeiro item acima, vamos agora estudar sobre as técnicas de levantamento de requisitos, detalhando seu conceito e suas respectivas vantagens e desvantagens, que podem ser utilizadas em conjunto pelo analista.

### Etnografia

A etnografia é caracterizada como uma técnica de observação, aonde o analista busca uma familiarização do cliente, seus valores, sua história. Ela pode ser utilizada para compreender os requisitos sociais e organizacionais, que facilitem compreensão da política organizacional e da sua cultura.

### Workshops



### Entrevistas

A entrevista também é uma das técnicas de levantamento de requisitos. Tradicionalmente mais simples de utilizar, produz bons resultados na fase inicial de obtenção de dados. Sobre a entrevista, deve-se considerar:

- O entrevistador dê margem ao entrevistado para expor as suas ideias.
- Ter um plano de entrevista para que seja mantido o foco no cerne do assunto principal.
- Evita que a entrevista fique longa, deixando o entrevistado cansado e não produzindo bons resultados.

### Questionários

Existem vários tipos de questionários que podem ser utilizados. Entre estes podemos listar: múltipla escolha, lista de verificação e questões com espaços em branco.

Um ambiente oportuno para o uso de questionário é quando há diversos grupos de usuários que podem estar em diversos locais diferentes do país. Neste caso, elaboram-se pesquisas específicas de acompanhamento com usuários selecionados, que a contribuição em potencial pareça mais importante, pois não seria prático entrevistar todas as pessoas em todos os locais.

### Questionários

- Identifique todos os destinatários que o receberão.
- Realize a distribuição junto com instruções detalhadas sobre seu preenchimento.
- Defina e informe o prazo para devolução do questionário.
- Documente o resultado da análise e consolidação das respostas dos participantes.
- Envie uma cópia com as informações levantadas para o participante, como sendo uma forma de agradecimento e consideração pelo tempo dedicado a pesquisa.

## **Brainstorming**

Brainstorming é uma técnica para geração de ideias. Uma ideia preliminar gerada serve como incentivo para que outras apareçam, sejam concordantes ou não, propiciando um aprimoramento e compartilhamento de todos os envolvidos. Pode ser estabelecida uma ou várias reuniões. O número de ideias geradas deve ser bem grande, pois quanto mais ideias forem propostas, maior será a chance de aparecerem boas ideias. Os participantes também devem ser encorajados a combinar ou enriquecer as ideias de outros e, para isso, é necessário que todas as ideias permaneçam visíveis a todos os participantes.

Uma vez definido que esta é a técnica mais apropriada para determinada situação, as próximas etapas necessárias para conduzir uma sessão de brainstorming estão elencadas nos balões em destaque.

- **Conteúdo 1 - Seleção dos participantes ou grupo de trabalho**

Estes devem ser selecionados em função direta com as contribuições que possam oferecer durante a sessão. É aconselhável sempre a presença de pessoas que estejam sempre bem informadas, sejam de diferentes grupos.

- **Conteúdo 2 - Prepara a sessão**

Duração e local do encontro, bem como o que será tratado.

- **Conteúdo 3 - Explicar a técnica e as regras a serem seguidas**

Definir os conceitos básicos de brainstorming e as regras a serem seguidas durante a sessão.

- **Conteúdo 4 - Gerar ou produzir uma boa quantidade de ideias**

Os participantes geram tantas ideias quantas forem exigidas pelos tópicos que estão sendo o objeto do brainstorming. Os participantes são convidados, um por vez, a dar uma única ideia. Se alguém tiver problema, passa a vez e espera a próxima rodada.

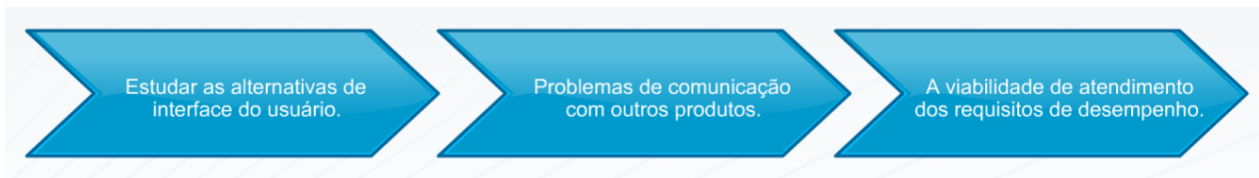
- **Conteúdo 5 - Analisar as ideias**

Revisar a produção de ideias, destacando as mais valiosas definidas pelo grupo e classificando-as com prioridades.

## **Prototipagem**

Protótipo tem por alvo explorar requisitos vinculados a um produto que possua aspectos críticos, implementando de maneira mais rápida um pequeno subconjunto de funcionalidades deste produto. O protótipo é aconselhado para:





As técnicas utilizadas na elaboração do protótipo são várias: interface de usuário, relatórios textuais, relatórios gráficos, entre outras. Um dos principais benefícios que podemos relacionar são as reduções dos riscos na construção do sistema, pois é possível detectar se o usuário chave já verificou o que o analista captou nos requisitos do produto.

São elementos para o sucesso na elaboração dos protótipos:

- Seleção do ambiente de prototipagem.
- Compreender os objetivos do protótipo por parte de todos os interessados no projeto.
- Focar em áreas que estejam com maior dificuldade na compreensão.
- Rapidez na construção.

## JAD

Por fim, não menos importante, vamos apresentar uma técnica de grande adesão pelos analistas a fim de realizar o levantamento de requisitos: JAD (Joint Application Design). É uma técnica destinada a principalmente promover cooperação, entendimento e trabalho em grupo entre os usuários desenvolvedores. Com a intenção de facilitar a criação de uma visão compartilhada do que o produto de software deve ser, ela ajuda os usuários e desenvolvedores a formular problemas e explorar soluções, no escopo do projeto a ser desenvolvido. Tal fato estabelece sentimento de envolvimento, posse e responsabilidade com o sucesso do produto.

Possui quatro princípios básicos:

**Dinâmica de grupo:** São realizadas reuniões com um líder experiente, analista, usuários e gerentes, para despertar a força e criatividade dos participantes. O resultado final será a determinação dos objetivos e requisitos do sistema.

**Uso de técnicas visuais:** Para aumentar a comunicação e o entendimento.

**Manutenção do processo organizado e racional:** O JAD emprega a análise top down e atividades bem definidas. Possibilita assim, a garantia de uma análise completa reduzindo as chances de falhas ou lacunas no projeto e cada nível de detalhe recebe a devida atenção.

**Utilização de documentação padrão:** Preenchida e assinada por todos os participantes. Este documento garante a qualidade esperada do projeto e promove a confiança dos participantes.

Possui um total de seis tipos de participantes – levando em consideração que nem todos participam de todas as fases. São eles:

|   |  |
|---|--|
| Líder da sessão:                        | É um facilitador dos encontros. Deve ser competente, com bom relacionamento pessoal e qualidades gerenciais de liderança.  |
| Engenheiro de requisitos:               | É um participante experiente nas questões técnicas, diretamente responsável pela produção dos documentos de saída das sessões JAD.   |
| Executor:                               | É o responsável pelo produto sendo construído.   |
| Representantes dos usuários:            | São pessoas na empresa que terão incumbência de utilizar o produto de software.  |
| Representantes de produtos de software: | São pessoas que estão familiarizadas com as capacidades dos produtos de software, capazes de mediar os usuários na compreensão entre o que é possível e razoável no sistema. |
| Especialista:                           | é a pessoa que pode fornecer informações detalhadas sobre um tópico específico.  |

## O que vem na próxima aula

- Documento de Requisitos de Software.
- Composição do Documento de Requisitos de Software.
- Utilidade e validade do Documento de Requisitos de Software.

## CONCLUSÃO

Nesta aula, você:

- Compreendeu o que é um stakeholders.
- Aprendeu sobre técnicas para levantamento de requisitos.
- Analisou diferentes estratégias para um bom levantamento de requisitos.