

# Unidade 2

## Tipos de Requisitos e as Técnicas de Levantamento

# Introdução da Unidade

**Olá, amigo(a) discente! Seja bem-vindo(a)!**

Já estudamos na unidade 01 sobre a importância dos requisitos no Processo de Desenvolvimento de Software. Desenvolver um software, sem conhecer os requisitos, é como construir uma casa sem saber o que o cliente deseja. Assim como uma casa não se faz somente com tijolos, quando desenvolvemos um software, também precisamos conhecer outros recursos.

Nesta unidade, vamos abordar sobre os tipos de requisitos existentes e as técnicas para efetuar o levantamento de requisitos.

## Objetivos

- Compreender os Requisitos Funcionais e Não-Funcionais;
- Compreender os tipos de leitores de requisitos;
- Explorar as Técnicas de Levantamento de Requisitos.

## Conteúdo Programático

**Aula 01** – Tipos de Requisitos e seus leitores

**Aula 02** – Técnicas de Levantamento de Requisitos



Quer **assistir às videoaulas** em seu celular? Basta apontar a câmera para os **QR Codes** distribuídos neste conteúdo.

Caso necessário, instale um aplicativo de leitura QR Code no celular e efetue o login na sua conta Gmail.

# Tipos de Requisitos e seus leitores

De acordo com o que foi estudado na Unidade 1, os requisitos estão presentes ao longo de todo o ciclo de vida de um software. Foi possível observar no ciclo de vida clássico, também conhecido como Cascata, que o levantamento de requisitos é efetuado no início do ciclo de vida. Observando os demais ciclos de vida como a prototipação e espiral, também foi possível observar que o levantamento de requisitos é um item indispensável para o desenvolvimento do software. Para o desenvolvimento do software, o bom entendimento dos requisitos é de extrema relevância para garantir a qualidade do mesmo.

Vale ressaltar que, os requisitos evoluem no decorrer do processo de desenvolvimento de software. Gerenciar a evolução dos requisitos é muito importante, estar ciente que os clientes nem sempre informam os requisitos no início do projeto e que tudo isso faz com que os requisitos possam mudar no decorrer do processo de desenvolvimento. Quando falamos em requisitos de software, não estamos falando apenas das funções do software e sim em outros recursos que envolvem o desenvolvimento desse software.

## Requisitos e seus leitores

Quando falamos em processo de engenharia de requisitos uma das falhas é distinguir os níveis de usuários no momento da descrição dos requisitos. Devemos considerar os requisitos descritos em dois níveis de detalhamento sendo eles para os requisitos de usuário e requisitos de sistemas.

Sommerville (2011), destaca como descrever os requisitos, de acordo com os níveis de usuários:

### 1. Requisitos de usuário

São declarações, em uma linguagem natural com diagramas, de quais serviços o sistema deverá fornecer a seus usuários e as restrições com as quais este deve operar.

### 2. Requisitos de sistema

São descrições mais detalhadas das funções, serviços e restrições operacionais do sistema de software. O documento de requisitos do sistema (às vezes chamado especificação funcional) deve definir exatamente o que deve ser implementado. Pode ser parte do contrato entre o comprador do sistema e os desenvolvedores de software.

A Figura 1, Sommerville (2011), ilustra as informações para diferentes tipos de leitor

Figura 1 - Requisitos de Usuário e de Sistema

**Definição de requisitos de usuário**

1. O MHC-PMS deve gerar relatórios gerenciais mensais que mostrem o custo dos medicamentos prescritos por cada clínica durante aquele mês.

**Especificação de requisitos de sistema**

1.1 No último dia útil de cada mês deve ser gerado um resumo dos medicamentos prescritos, seus custos e as prescrições de cada clínica.

1.2 Após as 17:30h do último dia útil do mês, o sistema deve gerar automaticamente o relatório para impressão.

1.3 Um relatório será criado para cada clínica, listando os nomes dos medicamentos, o número total de prescrições, o número de doses prescritas e o custo total dos medicamentos prescritos.

1.4 Se os medicamentos estão disponíveis em diferentes unidades de dosagem (por exemplo, 10 mg, 20 mg), devem ser criados relatórios separados para cada unidade.

1.5 O acesso aos relatórios de custos deve ser restrito a usuários autorizados por uma lista de controle de gerenciamento de acesso.

Fonte: Sommerville, 2011.

Os requisitos precisam ser escritos de forma diferente para diferentes tipos de leitores, como o descrito acima. O requisito dos usuários é descrito de forma com que usuário tenha um entendimento, porém sem muito detalhamento. Já a especificação de requisitos para quem vai desenvolver o sistema como Engenheiro de Sistema, Arquitetos de Sistema, devem ser escritos de forma mais detalhada pois é a partir desse detalhamento será implementado o sistema.



#### **Videoaula 1**

Utilize o QR Code para assistir!

Agora assista a videoaula sobre os diferentes tipos de leitores dos requisitos.



### **Requisitos Funcionais e Não-Funcionais**

Já vimos que existem tipos de leitores diferentes para os mesmos requisitos. Agora vamos estudar o que são os requisitos funcionais e não-funcionais. Conforme o RUP (IBM, 2007), existem vários tipos de requisitos. Um modo de categorizá-los é descrito usando o acrônimo **FURPS**, onde o **F** representa os Requisitos Funcionais e o **URPS** os requisitos não-funcionais. Veja algumas definições de requisitos funcionais e não funcionais.

#### **Requisitos Funcionais**

Descrevem o que ele deve fazer. Eles dependem do tipo de software a ser desenvolvido, de quem são seus possíveis usuários e da abordagem geral adotada pela organização ao escrever os requisitos. Quando expressos como requisitos de usuário, os requisitos funcionais são normalmente descritos de forma abstrata, para serem compreendidos pelos usuários do sistema. No entanto, requisitos de sistema funcionais mais específicos descrevem em detalhes as funções do sistema, suas entradas e saídas, exceções etc. (Sommerville, 2011). Requisitos funcionais do sistema variam de requisitos gerais, que abrangem o que o sistema deve fazer, até requisitos muito específicos, que refletem os sistemas e as formas de trabalho em uma organização. (Sommerville, 2011)

#### **Requisitos Não Funcionais**

Como o nome sugere, são requisitos que não estão diretamente relacionados com os serviços específicos oferecidos pelo sistema a seus usuários. Eles podem estar relacionados às propriedades emergentes do sistema, como confiabilidade, tempo de resposta e ocupação de área. Uma alternativa a esse cenário seria os requisitos que definem restrições sobre a implementação do sistema, como as capacidades dos dispositivos de E/S ou as representações de dados usadas nas interfaces com outros sistemas (Sommerville, 2011).

Os requisitos não-funcionais, como desempenho, proteção ou disponibilidade, normalmente especificam ou restringem as características do sistema como um todo. Requisitos não-funcionais são frequentemente mais críticos que requisitos funcionais individuais, os usuários do sistema podem, geralmente, encontrar maneiras de contornar uma função do sistema que realmente não atenda a suas necessidades. No entanto, deixar de atender a um requisito não-funcional pode significar a inutilização de todo o sistema. Por exemplo, se um sistema de aeronaves não cumprir seus requisitos de confiabilidade, não será certificado como um sistema seguro para operar. Se um sistema de controle embutido não atender aos requisitos de desempenho, as funções de controle não funcionarão corretamente (Sommerville, 2011).

Os requisitos funcionais especificam ações que um sistema deve ser capaz de executar, sem levar em consideração restrições físicas. Geralmente, isso é melhor descrito em um modelo de casos de uso e em casos de uso. Os requisitos funcionais especificam, portanto, o comportamento de entrada e saída de um sistema.



### Videoaula 2

Utilize o QR Code para assistir!

Agora, assista ao vídeo que aborda sobre os Requisitos Funcionais e Não-Funcionais.



Vários requisitos não são funcionais e descrevem apenas atributos do sistema ou atributos do ambiente do sistema. Os requisitos não funcionais são aqueles que dizem respeito a questões como as descritas abaixo.

### Usabilidade (URPS)

Os requisitos de usabilidade podem incluir subcategorias como:

- fatores humanos (consulte Conceitos: Design Centrado no Usuário)
- estética
- consistência na interface do usuário (consulte Diretrizes: Interface do Usuário)
- ajuda on-line e contextual
- assistentes e agentes
- documentação do usuário
- materiais de treinamento

### **Confiabilidade (URPS)**

Os requisitos de confiabilidade a serem considerados são:

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

### **Desempenho (URPS)**

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

- velocidade
- eficiência
- disponibilidade
- exatidão
- taxa de transferência
- tempo de resposta
- tempo de recuperação
- uso de recurso

### **Suportabilidade (FURPS)**

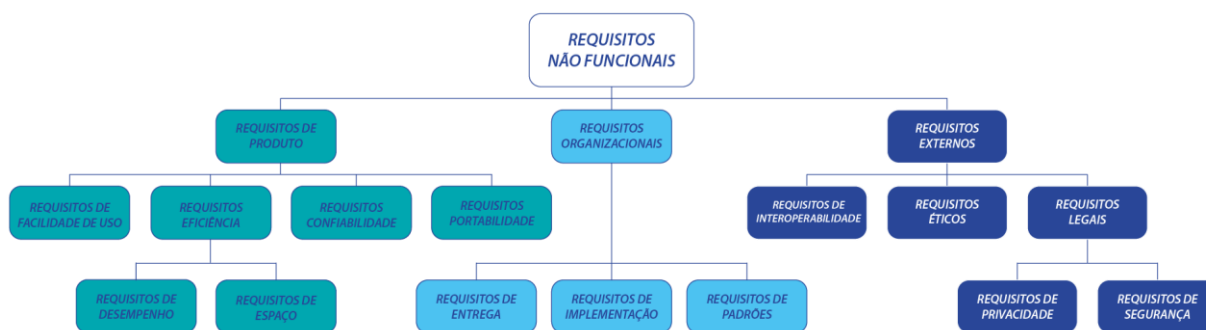
Os requisitos de suporte podem incluir:

- possibilidade de teste
- extensibilidade
- adaptabilidade
- manutenibilidade
- compatibilidade
- possibilidade de configuração
- possibilidade de serviço
- possibilidade de instalação
- possibilidade de localização (internacionalização)

Uma definição completa dos requisitos do software, dos casos de uso (requisitos Funcionais) e das Especificações Suplementares (Requisitos não-funcionais) pode ser reunida para definir uma Especificação de Requisitos de Software (SRS) para uma "característica" particular ou outros agrupamentos de subsistemas.

Na Figura 2 é possível observar que os requisitos não-funcionais podem ser provenientes das características requeridas para o software (requisitos de produto), da organização que desenvolve o software (requisitos organizacionais) ou de fontes externas.

Figura 2 - Tipos de requisitos não funcionais



Fonte: Sommerville, 2011.

Uma definição completa dos requisitos do software, dos casos de uso (requisitos Funcionais) e das Especificações Suplementares (Requisitos não-funcionais) pode ser reunida para definir uma Especificação de Requisitos de Software (SRS) para uma "característica" particular ou outros agrupamentos de subsistemas.

Na Figura 2 é possível observar que os requisitos não-funcionais podem ser provenientes das características requeridas para o software (requisitos de produto), da organização que desenvolve o software (requisitos organizacionais) ou de fontes externas.

Fonte: Sommerville, 2011.

### Requisitos de Produto

Esses requisitos especificam ou restringem o comportamento do software. Exemplos incluem os requisitos de desempenho quanto à rapidez com que o sistema deve executar e quanta memória ele requer, os requisitos de confiabilidade que estabelecem a taxa aceitável de falhas, os requisitos de proteção e os requisitos de usabilidade.

### Requisitos Organizacionais

Esses são os requisitos gerais de sistemas derivados das políticas e procedimentos da organização do cliente e do desenvolvedor. Exemplos incluem os requisitos do processo operacional, que definem como o sistema será usado, os requisitos do processo de desenvolvimento que especificam a linguagem de programação, o ambiente de desenvolvimento ou normas de processo a serem usadas, bem como os requisitos ambientais que especificam o ambiente operacional do sistema.

### Requisitos Externos

Esse tipo abrange todos os requisitos que derivam de fatores externos ao sistema e seu processo de desenvolvimento. Podem incluir requisitos reguladores, que definem o que deve



ser feito para que o sistema seja aprovado para uso, por um regulador, tal como um banco central; requisitos legais, que devem ser seguidos para garantir que o sistema opere dentro da lei; e requisitos éticos, que assegurem que o sistema será aceitável para seus usuários e o público em geral.



### Videoaula 3

Utilize o QR Code para assistir!

Agora, assista ao vídeo que aborda sobre os tipos de Requisitos Não Funcionais.



### Indicação de Leitura

Leia mais sobre os Requisitos Funcionais e Não Funcionais nos livros:

SOMMERVILLE, I. Engenharia de software. 9. ed. São Paulo: Pearson Education, 2011.

SIMÕES, G. S.; VAZQUEZ, C. E. **Engenharia de Requisitos**: software orientado ao negócio. Rio de Janeiro: Brasport, 2016.

Disponível em: <https://bit.ly/3rj3YpJ>. Acesso em: 15 jul. 2021.

Leia mais sobre os **requisitos não funcionais** acessando o RUP:

Disponível em: <https://bit.ly/3ihGT2N>. Acesso em: 15 jul. 2021.

Quer saber um pouco mais sobre os requisitos funcionais, acesse:

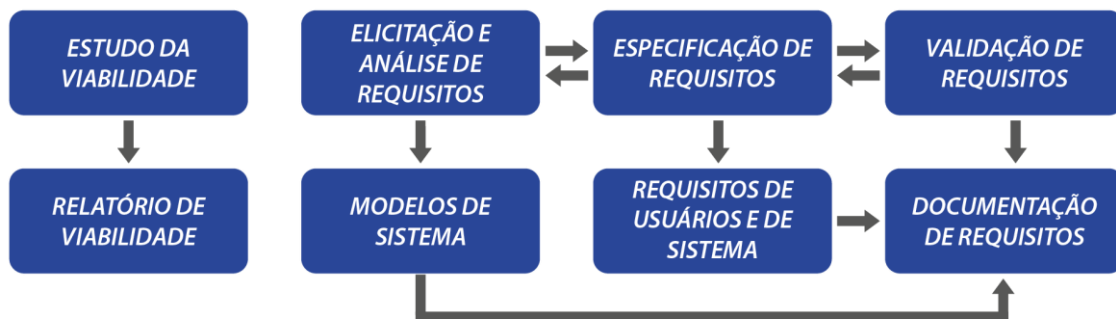
Disponível em: <https://bit.ly/3z6xoKB>. Acesso em: 15 jul. 2021.

# Técnicas para o Levantamento de Requisitos

Quando falamos sobre o desenvolvimento de software, na maioria das vezes o foco é o produto finalizado, esquecendo um ponto muito importante que é "como" extrair os requisitos de forma consistente. O método de extrair os requisitos para a fase de levantamento de informações pode ser explorado resultando em uma análise de requisitos mais consistente e realista (Chaves, 2021).

O levantamento e análise de requisitos é um processo iterativo, com uma contínua validação de uma atividade para outra, conforme exemplificado pela Figura 3.

Figura 3 - Processo de levantamento e análise de requisitos



Fonte: Sommerville, 2003.

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

- **Estudo de viabilidade.** É feita uma estimativa acerca da possibilidade de se satisfazerem as necessidades do usuário identificado usando-se tecnologias atuais de software e hardware. O estudo considera se o sistema proposto será rentável a partir de um ponto de vista de negócio e se ele pode ser desenvolvido no âmbito das atuais restrições orçamentais. Um estudo de viabilidade deve ser relativamente barato e rápido. O resultado deve informar a decisão de avançar ou não, com uma análise mais detalhada.
- **Levantamento e análise de requisitos.** Esse é o processo de derivação dos requisitos do sistema por meio da observação dos sistemas existentes, além de discussões com os potenciais usuários e compradores, análise de tarefas, entre outras etapas. Essa parte do processo pode envolver o desenvolvimento de um ou mais modelos de sistemas e protótipos, os quais nos ajudam a entender o sistema a ser especificado.

- **Especificação de requisitos.** É a atividade de traduzir as informações obtidas durante a atividade de análise em um documento que defina um conjunto de requisitos. Dois tipos de requisitos podem ser incluídos nesse documento. Requisitos do usuário são declarações abstratas dos requisitos do sistema para o cliente e usuário final do sistema; requisitos de sistema são uma descrição mais detalhada da funcionalidade a ser provida.
- **A validação de requisitos.** Essa atividade verifica os requisitos quanto ao realismo, consistência e completude. Durante esse processo os erros no documento de requisitos são inevitavelmente descobertos. Em seguida, o documento deve ser modificado para correção desses problemas. O problema de não saber especificar corretamente o que o sistema deverá fazer é muito antigo. Após mais de 30 anos da elaboração do relatório, a situação não é muito diferente. Algumas das razões para o baixo grau de satisfação dos usuários para os sistemas destacam-se: (Moraes, 2021).
- Na fase de levantamento de requisitos do projeto, onde não é utilizada uma técnica adequada para extrair os requisitos do sistema.
- A falha do analista em não descrever os requisitos do sistema de modo claro, sem ambiguidades, conciso e consistente com todos os aspectos significativos do sistema proposto.

Entre as dificuldades encontradas na fase de levantamento de requisitos estão:

- O usuário principal do sistema não sabe o que quer que o sistema faça ou sabe e não consegue transmitir para o analista;
- Requisitos identificados, mas que não são realistas e não identificam os requisitos similares informados por pessoas diferentes.



#### Videoaula 1

Utilize o QR Code para assistir!

Agora, assista ao vídeo que aborda sobre o processo de levantamento e análise de requisitos.



Identifica-se um levantamento de requisitos adequado por meio da boa definição do projeto, da efetividade do projeto, de informações necessárias a um perfeito diagnóstico e de soluções inteligentes. Quanto ao levantamento de requisitos inadequados, o resultado é um diagnóstico pobre com conclusões comprometidas, não identificação das causas dos problemas, custos elevados, prazos vencidos ou comprometedores, omissão de processos fundamentais e descréditos.

Existem várias técnicas para efetuar o levantamento de requisitos com o objetivo de superar as dificuldades desta fase. Todas as técnicas que vamos citar aqui, possuem seus próprios conceitos e formas de utilizá-las. Analisando as técnicas pode-se identificar que elas não são completas. Com isso, o ideal é que seja utilizado uma combinação de técnicas, fazendo com que as lacunas sejam preenchidas. Vale ressaltar que o método a ser utilizado depende do tipo de projeto a ser realizado.

As técnicas de levantamento de requisitos têm por objetivo superar as dificuldades relativas a esta fase. Todas as técnicas possuem um conceito próprio e suas respectivas vantagens e desvantagens, que podem ser utilizadas em conjunto pelo analista.

Segue abaixo algumas técnicas de levantamento, como Levantamento orientado a pontos de vista, Etnografia, Workshops, Prototipagem, Entrevistas e Questionários. (Sommerville, 2003).

### **Levantamento orientado a pontos de vista**

Para qualquer sistema, de tamanho médio ou grande, normalmente há diferentes tipos de usuário final. Muitos stakeholders têm algum tipo de interesse nos requisitos do sistema. Por esse motivo, mesmo para um sistema relativamente simples, existem muitos pontos de vista diferentes que devem ser considerados. Os diferentes pontos de vista a respeito de um problema ‘veem’ o problema de modos diferentes. Contudo, suas perspectivas não são inteiramente independentes, mas em geral apresentam alguma duplicidade, de modo que apresentam requisitos comuns.

As abordagens orientadas a ponto de vista, na engenharia de requisitos, reconhecem esses diferentes pontos de vista e os utilizam para estruturar e organizar o processo de levantamento e os próprios requisitos. Uma importante capacidade da análise orientada a pontos de vista é que ela reconhece a existência de várias perspectivas e oferece um *framework* para descobrir conflitos nos requisitos propostos por diferentes *stakeholders*.

O método VORD (*viewpoint-oriented requirements definition* – definição de requisitos orientada a ponto de vista) foi projetado como um framework orientado a serviço para o levantamento e análise de requisitos. A primeira etapa da análise de ponto de vista é identificar os possíveis pontos de vista. Nessa etapa os analistas se reúnem com os *stakeholders* e utilizam a abordagem de brainstorming para identificar os serviços em potencial e as entidades que interagem com o sistema.

A segunda etapa é a estruturação de pontos de vista, que envolve agrupar pontos de vista relacionados, segundo uma hierarquia. Serviços comuns estão localizados nos níveis mais altos da hierarquia e herdados por pontos de vista de nível inferior. A etapa de documentação do ponto de vista tem por objetivo refinar a descrição dos pontos de vista e serviços identificados.

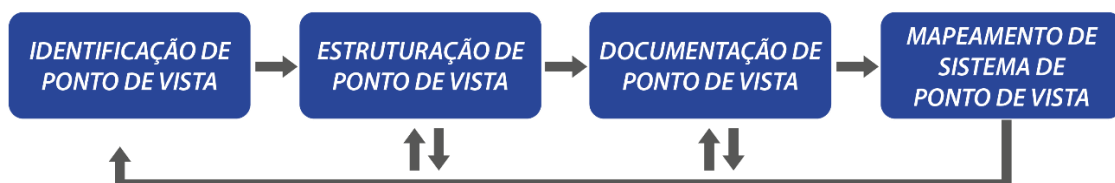
O mapeamento de sistema conforme ponto de vista envolve identificar objetos em um projeto orientado a objetos, utilizando as informações de serviço que estão encapsuladas nos pontos de vista.



O mapeamento de sistema conforme ponto de vista envolve identificar objetos em um projeto orientado a objetos, utilizando as informações de serviço que estão encapsuladas nos pontos de vista.

A Figura 4 exemplifica a técnica de levantamento orientado a ponto de vista.

Figura 4 - Método VORD



Fonte: Sommerville, 2003.

## Etnografia

A etnografia é uma técnica de observação que pode ser utilizada para compreender os requisitos sociais e organizacionais, ou seja, entender a política organizacional bem como a cultura de trabalho com objetivo de familiarizar-se com o sistema e sua história. Os cientistas sociais e antropólogos usam técnicas de observação para desenvolver um entendimento completo e detalhado de culturas particulares.

Nesta técnica o analista se insere no ambiente de trabalho em que o sistema será utilizado. O trabalho diário é observado e são anotadas as tarefas reais em que o sistema será utilizado. O principal objetivo da etnografia é que ela ajuda a descobrir requisitos de sistema implícitos, que refletem os processos reais, em vez de os processos formais, onde as pessoas estão envolvidas.



Etnografia é particularmente eficaz na descoberta de dois tipos de requisitos:

- Os requisitos derivados da maneira como as pessoas realmente trabalham, em vez da maneira pelas quais as definições de processo dizem como elas deveriam trabalhar;
- Os requisitos derivados da cooperação e conscientização das atividades de outras pessoas.

Alguns itens importantes que devem ser executados antes, durante e depois do estudo de observação:

- Antes, é necessário identificar as áreas de usuário a serem observadas, obter a aprovação das gerências apropriadas para executar as observações, obter os nomes e funções das pessoas chave que estão envolvidas no estudo de observação e explicar a finalidade do estudo;
- Durante, é necessário familiarizar-se com o local de trabalho que está sendo observado. Para isso é preciso observar os agrupamentos organizacionais atuais, as facilidades manuais e automatizadas, coletar amostras de documentos e procedimentos escritos que são usados em cada processo específico que está sendo observado e acumular informações estatísticas a respeito das tarefas, como: frequência que ocorrem, estimativas de volumes, tempo de duração para cada pessoa que está sendo observada. Além de observar as operações normais de negócios acima é importante observar as exceções;
- Depois, é necessário documentar as descobertas resultantes das observações feitas. Para consolidar o resultado é preciso rever os resultados com as pessoas observadas e/ou com seus superiores.

A análise de observação tem algumas desvantagens como, consumir bastante tempo e o analista ser induzido a erros em suas observações. Mas em geral a técnica de observação é muito útil e frequentemente usada para complementar descobertas obtidas por outras técnicas.

## Workshops

Trata-se de uma técnica de elicitação em grupo usada em uma reunião estruturada. Devem fazer parte do grupo uma equipe de analistas e uma seleção dos *stakeholders* que melhor representam a organização e o contexto em que o sistema será usado, obtendo assim um conjunto de requisitos bem definidos.

Ao contrário das reuniões, onde existe pouca interação entre todos os elementos presentes, o *workshop* tem o objetivo de acionar o trabalho em equipe. Há um facilitador neutro cujo papel é conduzir a *workshop* e promover a discussão entre os vários mediadores. As tomadas de decisão são baseadas em processos bem definidos e com o objetivo de obter um processo de negociação, mediado pelo facilitador.

Uma técnica utilizada em *workshops* é o *brainstorming*. Após os *workshops* serão produzidas documentações que refletem os requisitos e decisões tomadas sobre o sistema a ser desenvolvido.

Alguns aspectos importantes a serem considerados: a postura do condutor do seminário deve ser de mediador e observador, a convocação deve possuir dia, hora, local, horário de início e de término, assunto a ser discutido e a documentação do seminário.



### Videoaula 2

Utilize o QR Code para assistir!

Agora, assista ao vídeo que aborda sobre o processo de levantamento e análise de requisitos.



## Prototipagem

Protótipo tem por objetivo explorar aspectos críticos dos requisitos de um produto, implementando de forma rápida um pequeno subconjunto de funcionalidades deste produto. O protótipo é indicado para estudar as alternativas de interface do usuário, problemas de comunicação com outros produtos e a viabilidade de atendimento dos requisitos de desempenho. As técnicas utilizadas na elaboração do protótipo são várias: interface de usuário, relatórios textuais, relatórios gráficos, entre outras.



Alguns dos benefícios do protótipo são as reduções dos riscos na construção do sistema, pois o usuário chave já verificou o que o analista captou nos requisitos do produto. Para ter sucesso na elaboração dos protótipos é necessária a escolha do ambiente de prototipagem, o entendimento dos objetivos do protótipo por parte de todos os interessados no projeto, a focalização em áreas menos compreendidas e a rapidez na construção.

## **Entrevistas**

A entrevista é uma das técnicas tradicionais mais simples de utilizar e que produz bons resultados na fase inicial de obtenção de dados. Convém que o entrevistador dê margem ao entrevistado para expor as suas ideias. É necessário ter um plano de entrevista para que não haja dispersão do assunto principal e a entrevista fique longa, deixando o entrevistado cansado e não produzindo bons resultados.

As seguintes diretrizes podem ser de grande auxílio na direção de entrevistas bem sucedidas com o usuário: desenvolver um plano geral de entrevistas, certificar-se da autorização para falar com os usuários, planejar a entrevista para fazer uso eficiente do tempo, utilizar ferramentas automatizadas que sejam adequadas, tentar descobrir que informação o usuário está mais interessado e usar um estilo adequado ao entrevistar.

Para planejar a entrevista é necessário que antes dela sejam coletados e estudados todos os dados pertinentes à discussão, como formulários, relatórios, documentos e outros. Dessa forma, o analista estará bem contextualizado e terá mais produtividade nos assuntos a serem discutidos na entrevista.

É importante determinar um escopo relativamente limitado, focalizando uma pequena parte do sistema para que a reunião não se estenda por mais de uma hora. O usuário tem dificuldade de concentração em reuniões muito longas, por isso, é importante focalizar a reunião no escopo definido.

Após a entrevista é necessário validar se o que foi documentado pelo analista está de acordo com a necessidade do usuário, que o usuário não mudou de opinião e que o usuário entende a notação ou representação gráfica de suas informações.

A atitude do analista em relação à entrevista é determinar seu fracasso ou sucesso. Uma entrevista não é uma competição, deve-se evitar o uso excessivo de termos técnicos e não conduzir a entrevista em uma tentativa de persuasão. O modo como o analista fala não deve ser muito alto, nem muito baixo, tampouco indiretamente, ou seja, utilizar os termos: ele disse isso ou aquilo na reunião para o outro entrevistado.

O melhor modo para agir seria, por exemplo, dizer: O João vê a solução para o projeto dessa forma. E o senhor André, qual é a sua opinião? Em uma entrevista o analista nunca deve criticar a credibilidade do entrevistado. O analista deve ter em mente que o entrevistado é o perito no assunto e fornecerá as informações necessárias ao sistema.

Para elaborar perguntas detalhadas é necessário solicitar que o usuário:

- Explique o relacionamento entre o que está em discussão e as demais partes do sistema;
- Descreva o ponto de vista de outros usuários em relação ao item que esteja sendo discutido;
- Descreva informalmente a narrativa do item em que o analista deseja obter informações;



- Perguntar ao usuário se o item em discussão depende para a sua existência de alguma outra coisa, para assim poder juntar os requisitos comuns do sistema, formando assim um escopo conciso.

Pode-se utilizar a confirmação, para tanto o analista deve dizer ao usuário o que acha que ouviu ele dizer. Neste caso, o analista deve utilizar as suas próprias palavras em lugar das do entrevistado e solicitar ao entrevistado confirmação do que foi dito.

### **Questionários**

O uso de questionário é indicado, por exemplo, 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.

Existem vários tipos de questionários que podem ser utilizados. Entre estes podemos listar: múltipla escolha, lista de verificação e questões com espaços em branco. O questionário deve ser desenvolvido de forma a minimizar o tempo gasto em sua resposta.

Na fase de preparação do questionário deve ser indicado o tipo de informação que se deseja obter. Assim que os requisitos forem definidos o analista deve elaborar o questionário com questões de forma simples, clara e concisa, deixar espaço suficiente para as respostas que forem descritivas e agrupar as questões de tópicos específicos em um conjunto com um título especial. O questionário deve ser acompanhado por uma carta explicativa, redigida por um alto executivo, para enfatizar a importância dessa pesquisa para a organização.

Deve ser desenvolvido um controle que identifique todas as pessoas que receberam os questionários. A distribuição deve ocorrer junto com instruções detalhadas sobre como preenchê-lo e ser indicado claramente o prazo para devolução do questionário. Ao analisar as respostas dos participantes é feita uma consolidação das informações fornecidas no questionário, documentando as principais descobertas e enviando uma cópia com estas informações para o participante como forma de consideração pelo tempo dedicado à pesquisa.

### **Brainstorming**

*Brainstorming* é uma técnica para geração de ideias. Ela consiste em uma ou várias reuniões que permitem que as pessoas sugiram e explorem ideias.

As principais etapas necessárias para conduzir uma sessão de *brainstorming* são:

- Seleção dos participantes: Os participantes devem ser selecionados em função das contribuições diretas que possam dar durante a sessão. A presença de pessoas bem informadas, vindas de diferentes grupos garantirá uma boa representação;
- Explicar a técnica e as regras a serem seguidas: O líder da sessão explica os conceitos básicos de *brainstorming* e as regras a serem seguidas durante a sessão;
- Produzir uma boa quantidade de ideias: Os participantes geram tantas ideias quantas forem exigidas pelos tópicos que estão sendo objeto do *brainstorming*. Os participantes são convidados, um por vez, a dar uma única idéia. Se alguém tiver problema, passa a vez e espera a próxima rodada.

No *brainstorming* as ideias que a princípio pareçam não convencionais, são encorajadas, pois elas frequentemente estimulam os participantes, o que pode levar a soluções criativas para o problema. 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.

Nesta técnica é designada uma pessoa para registrar todas as ideias em uma lousa branca ou em papel. À medida que cada folha de papel é preenchida, ela é colocada de forma que todos os participantes possam vê-la.

Analisar as ideias é a fase final do *brainstorming*. Nessa fase é realizada uma revisão das ideias, uma de cada vez. As consideradas valiosas pelo grupo são mantidas e classificadas em ordem de prioridade.

## **JAD**

JAD (*Joint Application Design*) é uma técnica para promover cooperação, entendimento e trabalho em grupo entre os usuários desenvolvedores. O JAD facilita a criação de uma visão compartilhada do que o produto de software deve ser. Através da sua utilização, os desenvolvedores ajudam os usuários a formular problemas e explorar soluções. Dessa forma, os usuários ganham um sentimento de envolvimento, posse e responsabilidade com o sucesso do produto.

A técnica JAD tem 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.

A técnica JAD é composta de duas etapas principais: planejamento, que tem por objetivo elicitar e especificar os requisitos; e projeto, em que se lida com o projeto de software.

Cada etapa consiste em três fases: adaptação, sessão e finalização. A fase de adaptação consiste na preparação para a sessão, ou seja, organizar a equipe, adaptar o processo JAD ao produto a ser construído e preparar o material. Na fase de sessão é realizado um ou mais

encontros estruturados, envolvendo desenvolvedores e usuários onde os requisitos são desenvolvidos e documentados. A fase de finalização tem por objetivo converter a informação da fase de sessão em sua forma final (um documento de especificação de requisitos).

O conceito do JAD de abordagem e dinâmica de grupo poderá ser utilizado para diversas finalidades, como: planejamento de atividades técnicas para um grande projeto, discussão do escopo e objetivos de um projeto e estimativa da quantidade de horas necessárias para desenvolver sistemas grandes e complexos.

A maioria das técnicas JAD funcionam melhor em projetos pequenos ou médios. Para um sistema grande e complexo podem ser usadas múltiplas sessões JAD para acelerar a definição dos requisitos do sistema.

#### Indicação de Leitura

Leia mais sobre as técnicas de levantamento de requisitos acessando:

CHAVES, Ana Maria Siqueira *et al.* **Técnicas de extração de requisitos.**

Disponível em: <https://www.tiespecialistas.com.br/tecnicas-de-extracao-de-requisitos/>.

Acesso em: 30 maio 2021.

MORAES, Janaína Bedani Dixon. **Técnicas para levantamento de Requisitos.**

Disponível em: <https://www.devmedia.com.br/tecnicas-para-levantamento-de-requisitos/9151>. Acesso em: 30 maio 2021.



#### Videoaula 3

Utilize o QR Code para assistir!

Agora, assista ao vídeo que aborda as técnicas de levantamento de Requisito.



#### Fórum Avaliativo

##### **Prezado(a) aluno(a)!**

Agora convido você para realizar a primeira atividade avaliativa da disciplina: **Fórum de Discussões.**

Para participar, você deverá clicar em “**Fóruns**” no “**Menu Lateral**” e acessar.

**Lembre-se:** Após a data estipulada o Fórum será encerrado e não será mais permitido participar.

Sua contribuição é muito importante. Bons estudos!

## Videoaulas

### Prezado(a) aluno(a)!

Você também poderá encontrar todas as videoaulas, clicando em “**Módulos**” no “**Menu Lateral**” e acessar a página de vídeos.

## Encerramento

Caros alunos, nesta unidade foi possível compreender a importância de conhecer os tipos de requisitos e seus leitores. Saber que existem tipos diferentes de requisitos, nos deixa preparados para efetuar o levantamento do mesmo.

Para fazer o levantamento, vimos que existem diversos tipos de técnicas, sendo que as mesmas podem ser utilizadas de forma mesclada, uma complementando a outra.

É importante saber que levantar os requisitos da maneira correta evita a possibilidade de erros no desenvolvimento do projeto. Conhecer quem vai ler os requisitos, é importante para saber como vamos documentá-los.

Na próxima unidade vamos estudar sobre a documentação dos requisitos levantados. Até a próxima.

## Referências

CHAVES, A. M. S. *et al.* **Técnicas de extração de requisitos.**

Disponível em: <https://www.tiespecialistas.com.br/tecnicas-de-extracao-de-requisitos/>.

Acesso em: 15 jul. 2021.

GUEDES, G.T. A. **UML 2 – Uma Abordagem Prática.** 3. ed. São Paulo: Novatec, 2018.

IBM. **RATIONAL SOFTWARE CORPORATION.** IBM Rational Unified Process v2007. IBM, 2007.

MORAES, J. B. D. **Técnicas para levantamento de Requisitos.**

Disponível em: <https://www.devmedia.com.br/tecnicas-para-levantamento-de-requisitos/9151>. Acesso em: 15 jul. 2021.

PAULA FILHO, W. **Engenharia de Software: fundamentos, métodos e padrões.** 3. ed.

SIMÕES, Guilherme Siqueira; VAZQUEZ, Carlos Eduardo. **Engenharia de Requisitos: software orientado ao negócio.** Rio de Janeiro: Brasport, 2016. 372 p.

SOMMERVILLE, I. **Engenharia de software.** 6. ed. São Paulo: Pearson Education, 2003.

SOMMERVILLE, I. **Engenharia de software.** 9. ed. São Paulo: Pearson Education, 2011.

Esperamos que este guia o tenha ajudado compreender a organização e o funcionamento de seu curso. Outras questões importantes relacionadas ao curso serão disponibilizadas pela coordenação.

Grande abraço e sucesso!

