

# 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...



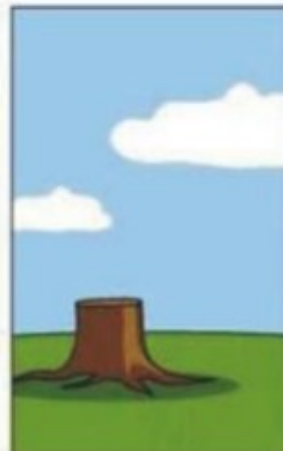
Como o projeto foi documentado...



Que funcionalidades foram instaladas...



Como o cliente foi cobrado...



Como foi mantido...



O que o cliente realmente queria...

# **Documento de requisitos de software**

# Documento de requisitos de software

---

O documento de requisitos de software, às vezes chamado Especificação de Requisitos de Software (SRS — do inglês *Software Requirements Specification*), é uma declaração oficial de o que os desenvolvedores do sistema devem implementar.

Deve incluir tanto os requisitos de usuário para um sistema quanto uma especificação detalhada dos requisitos de sistema

# Dica sobre análise de requisitos

---

O analista de requisitos, especifica com o Documento de requisitos de software **O QUE** precisa ser desenvolvido, e não no **COMO**.

O QUE – Foco da análise de requisitos.

COMO – Foco no projeto e implementação

# Documento de requisitos de software

---

Documentos de requisitos são essenciais quando um contratante externo está desenvolvendo o sistema de software.

Entretanto, os métodos ágeis de desenvolvimento argumentam que os requisitos mudam tão rapidamente que um documento de requisitos já está ultrapassado assim que termina de ser escrito.

# Documento de requisitos de software

---

Em vez de um documento formal, abordagens como a Extreme Programming (BECK, 1999) coletam os requisitos de usuário de forma incremental e escrevem-nos em cartões como histórias de usuário.

O usuário então prioriza os requisitos para implementação no próximo incremento do sistema.

# Documento de requisitos de software

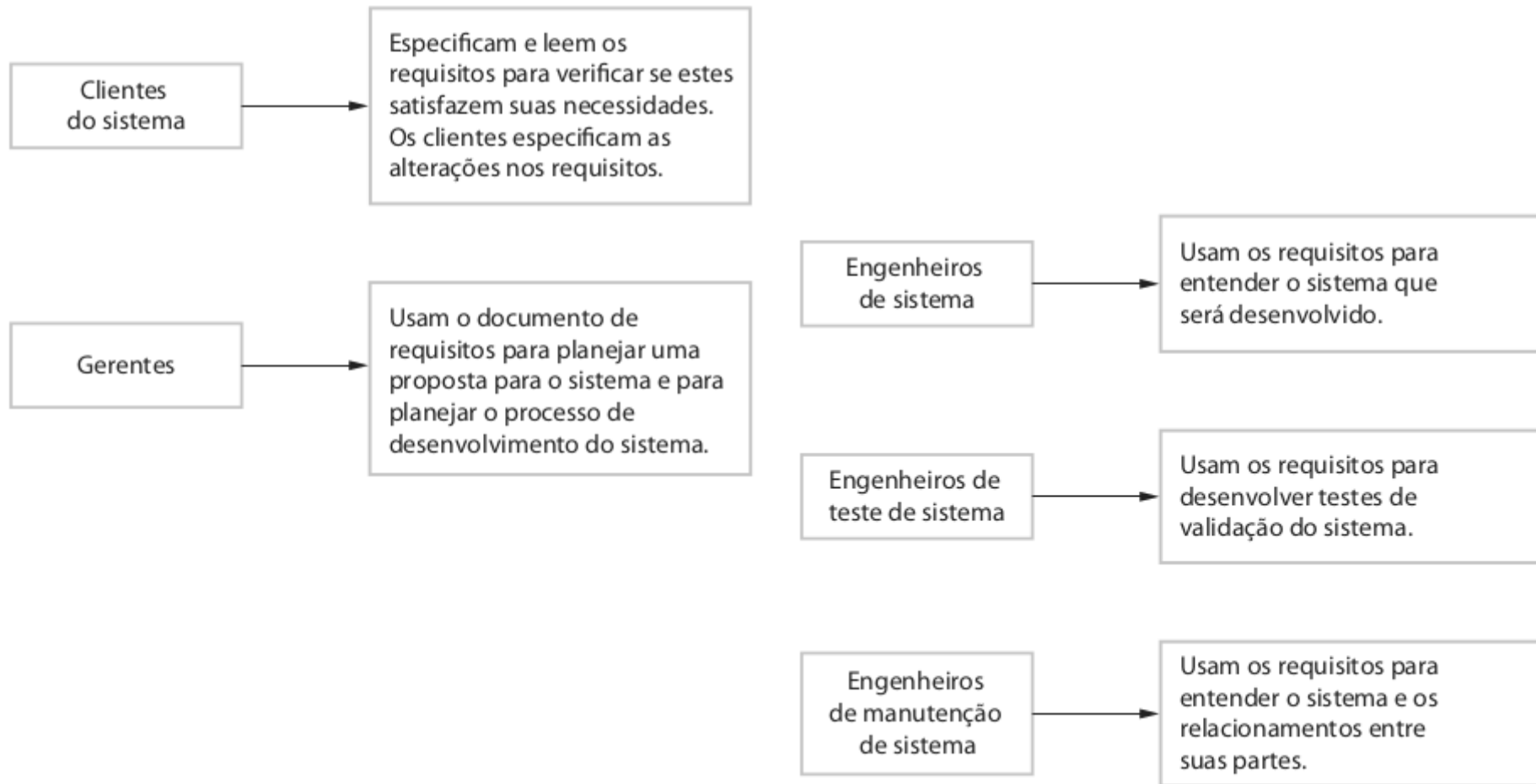
---



O documento de requisitos tem um conjunto diversificado de usuários, que vão desde a alta administração da organização que está pagando pelo sistema até os engenheiros responsáveis pelo desenvolvimento do software.



# Documento de requisitos de software



# Documento de requisitos de software

---

O nível de detalhes que você deve incluir em um documento de requisitos depende do tipo de sistema em desenvolvimento e o processo usado.

Os sistemas críticos precisam ter requisitos detalhados, porque a segurança e a proteção devem ser analisadas em detalhes.

# Documento de requisitos de software

---

Quando o sistema está sendo desenvolvido por uma companhia separada (por exemplo, através de *outsourcing*), as especificações de sistema devem ser detalhadas e precisas.

Se um processo interno de desenvolvimento iterativo é usado, o documento de requisitos pode ser muito menos detalhado e quaisquer ambiguidades podem ser resolvidas durante o desenvolvimento do sistema.

# Documento de requisitos de software

---

A Tabela 4.2 mostra uma possível organização de um documento de requisitos baseada em uma norma IEEE para documentos de requisitos (IEEE, 1998).

Essa é uma norma genérica que pode ser adaptada para usos específicos.

# Documento de requisitos de software

**Tabela 4.2**

A estrutura de um documento de requisitos.

Capítulo	Descrição
Prefácio	Deve definir os possíveis leitores do documento e descrever seu histórico de versões, incluindo uma justificativa para a criação de uma nova versão e um resumo das mudanças feitas em cada versão.
Introdução	Deve descrever a necessidade para o sistema. Deve descrever brevemente as funções do sistema e explicar como ele vai funcionar com outros sistemas. Também deve descrever como o sistema atende aos objetivos globais de negócio ou estratégicos da organização que encomendou o software.
Glossário	Deve definir os termos técnicos usados no documento. Você não deve fazer suposições sobre a experiência ou o conhecimento do leitor.
Definição de requisitos de usuário	Deve descrever os serviços fornecidos ao usuário. Os requisitos não funcionais de sistema também devem ser descritos nessa seção. Essa descrição pode usar a linguagem natural, diagramas ou outras notações compreensíveis para os clientes. Normas de produto e processos que devem ser seguidos devem ser especificados.

# Documento de requisitos de software

Arquitetura do sistema	Deve apresentar uma visão geral em alto nível da arquitetura do sistema previsto, mostrando a distribuição de funções entre os módulos do sistema. Componentes de arquitetura que são reusados devem ser destacados.
Especificação de requisitos do sistema	Deve descrever em detalhes os requisitos funcionais e não funcionais. Se necessário, também podem ser adicionados mais detalhes aos requisitos não funcionais. Interfaces com outros sistemas podem ser definidas.
Modelos do sistema	Pode incluir modelos gráficos do sistema que mostram os relacionamentos entre os componentes do sistema, o sistema e seu ambiente. Exemplos de possíveis modelos são modelos de objetos, modelos de fluxo de dados ou modelos semânticos de dados.
Evolução do sistema	Deve descrever os pressupostos fundamentais em que o sistema se baseia, bem como quaisquer mudanças previstas, em decorrência da evolução de hardware, de mudanças nas necessidades do usuário etc. Essa seção é útil para projetistas de sistema, pois pode ajudá-los a evitar decisões capazes de restringir possíveis mudanças futuras no sistema.
Apêndices	Deve fornecer informações detalhadas e específicas relacionadas à aplicação em desenvolvimento, além de descrições de hardware e banco de dados, por exemplo. Os requisitos de hardware definem as configurações mínimas ideais para o sistema. Requisitos de banco de dados definem a organização lógica dos dados usados pelo sistema e os relacionamentos entre esses dados.
Índice	Vários índices podem ser incluídos no documento. Pode haver, além de um índice alfabético normal, um índice de diagramas, de funções, entre outros pertinentes.

A linguagem natural tem sido usada para escrever os requisitos para o software.

**Tabela 4.3**

Formas de escrever uma especificação de requisitos de sistema.

Notação	Descrição
Sentenças em linguagem natural	Os requisitos são escritos em frases numeradas em linguagem natural. Cada frase deve expressar um requisito.
Linguagem natural estruturada	Os requisitos são escritos em linguagem natural em um formulário padrão ou <i>template</i> . Cada campo fornece informações sobre um aspecto do requisito.
Linguagem de descrição de projeto	Essa abordagem usa uma linguagem como de programação, mas com características mais abstratas, para especificar os requisitos, definindo um modelo operacional do sistema. Essa abordagem é pouco usada atualmente, embora possa ser útil para as especificações de interface.
Notações gráficas	Para definição dos requisitos funcionais para o sistema são usados modelos gráficos, suplementados por anotações de texto; diagramas de caso de uso e de sequência da UML são comumente usados.
Especificações matemáticas	Essas notações são baseadas em conceitos matemáticos, como máquinas de estado finito ou conjuntos. Embora essas especificações inequívocas possam reduzir a ambiguidade de um documento de requisitos, a maioria dos clientes não entende uma especificação formal. Eles não podem verificar que elas representam o que eles querem e são relutantes em aceitá-las como um contrato de sistema.

---

**Quadro 4.3** Uma especificação estruturada de um requisito para uma bomba de insulina.**Bomba de insulina/Software de controle/SRS/3.3.2**

<b>Função</b>	Calcula doses de insulina: nível seguro de açúcar.
<b>Descrição</b>	Calcula a dose de insulina a ser fornecida quando o nível de açúcar está na zona de segurança entre três e sete unidades.
<b>Entradas</b>	Leitura atual de açúcar ( $r_2$ ), duas leituras anteriores ( $r_0$ e $r_1$ ).
<b>Fonte</b>	Leitura atual da taxa de açúcar pelo sensor. Outras leituras da memória.
<b>Saídas</b>	CompDose — a dose de insulina a ser fornecida.
<b>Destino</b>	Loop principal de controle.
<b>Ação</b>	CompDose é zero se o nível de açúcar está estável ou em queda ou se o nível está aumentando, mas a taxa de aumento está diminuindo. Se o nível está aumentando e a taxa de aumento está aumentando, então CompDose é calculado dividindo-se a diferença entre o nível atual de açúcar e o nível anterior por quatro e arredondando-se o resultado. Se o resultado é arredondado para zero, então CompDose é definida como a dose mínima que pode ser fornecida.
<b>Requisitos</b>	Duas leituras anteriores, de modo que a taxa de variação do nível de açúcar pode ser calculada.
<b>Pré-condição</b>	O reservatório de insulina contém, no mínimo, o máximo de dose única permitida de insulina.
<b>Pós-condições</b>	$r_0$ é substituída por $r_1$ e $r_1$ é substituída por $r_2$ .
<b>Efeitos colaterais</b>	Nenhum.



**Tabela 4.4** Especificação tabular de processamento para uma bomba de insulina.

Condição	Ação
Nível de açúcar diminuindo ( $r_2 < r_1$ )	CompDose = 0
Nível de açúcar estável ( $r_2 = r_1$ )	CompDose = 0
Nível de açúcar aumentando e a taxa de aumento decrescente [ $(r_2 - r_1) < (r_1 - r_0)$ ]	CompDose = 0
Nível de açúcar aumentando e a taxa de aumento estável ou crescente [ $(r_2 - r_1) \geq (r_1 - r_0)$ ]	CompDose = arredondar $[(r_2 - r_1) / 4]$ . Se o resultado arredondado = 0, então CompDose = MinimumDose

- Descrever o quê o sistema deve fazer, em acordo com o cliente e usuários
- Descrever como gerenciar escopo e mudanças de requisitos
- Delimitar o sistema e prover uma base para o planejamento das iterações
- Definir a interface com o usuário

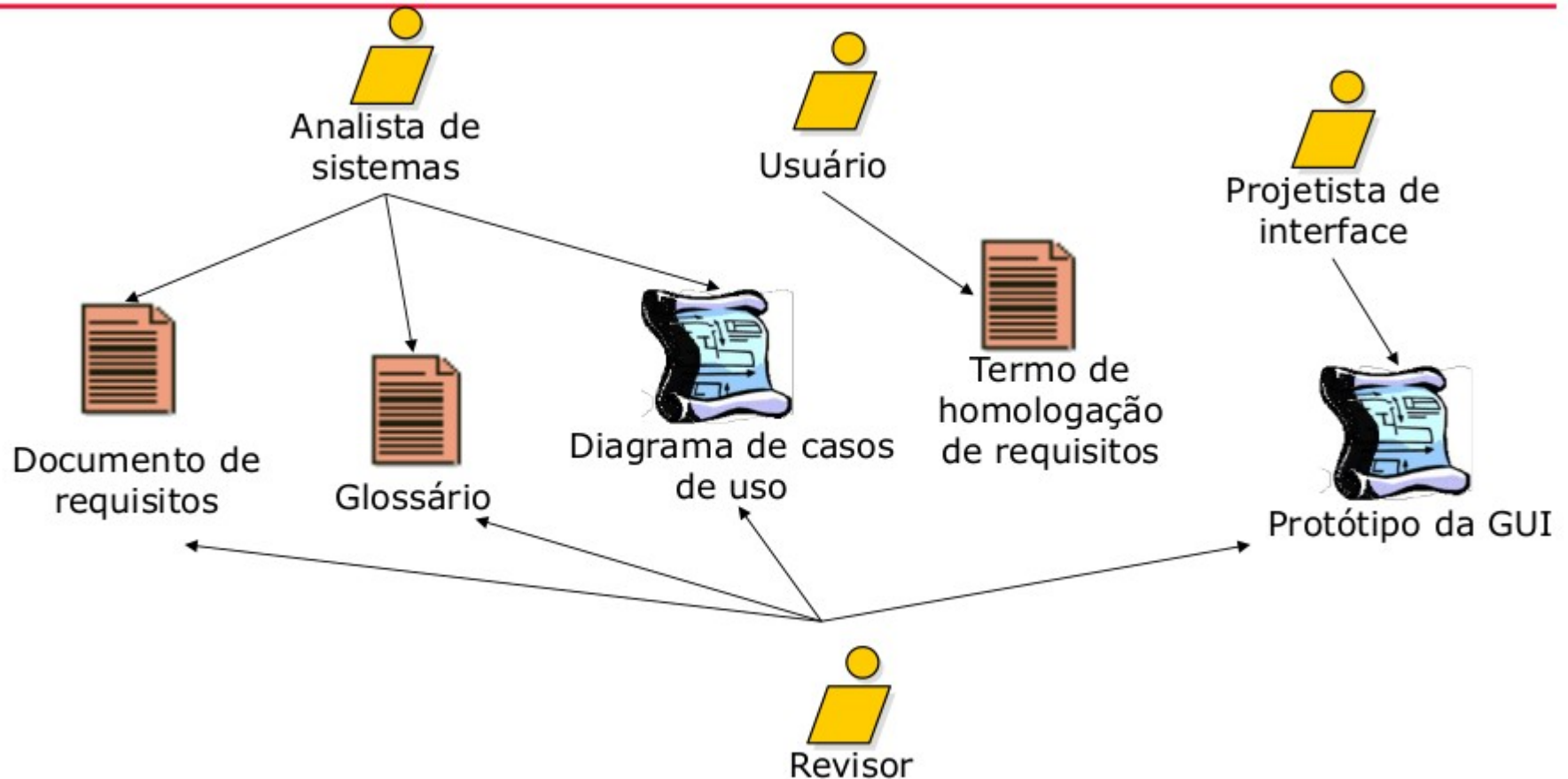
# Devemos ter em mente...

---

- O sistema deve prover valor ao cliente e usuários
- Os requisitos precisam ser definidos na direção correta
- Os clientes precisam entender o resultado da captura de requisitos

- Define termos importantes usados no projeto
- É importante para garantir que os conceitos envolvidos são interpretados da mesma forma por todos os membros da equipe

# Responsáveis nos fluxos de requisitos



Fonte: Jaelson Castro, 2016

# **Prática com Documento de requisitos de software**

# **Casos de Uso**

Os casos de uso são uma técnica de descoberta de requisitos introduzida inicialmente no método Objectory (JACOBSON et al., 1993). Eles já se tornaram uma característica fundamental da linguagem de modelagem unificada (UML — do inglês *unified modeling language*).



Em sua forma mais simples, um caso de uso identifica os atores envolvidos em uma interação e dá nome ao tipo de interação. Essa é, então, suplementada por informações adicionais que descrevem a interação com o sistema.

A informação adicional pode ser uma descrição textual ou um ou mais modelos gráficos, como diagrama de sequência ou de estados da UML.

Os casos de uso são documentados por um diagrama de casos de uso de alto nível.

O conjunto de casos de uso representa todas as possíveis interações que serão descritas nos requisitos de sistema.

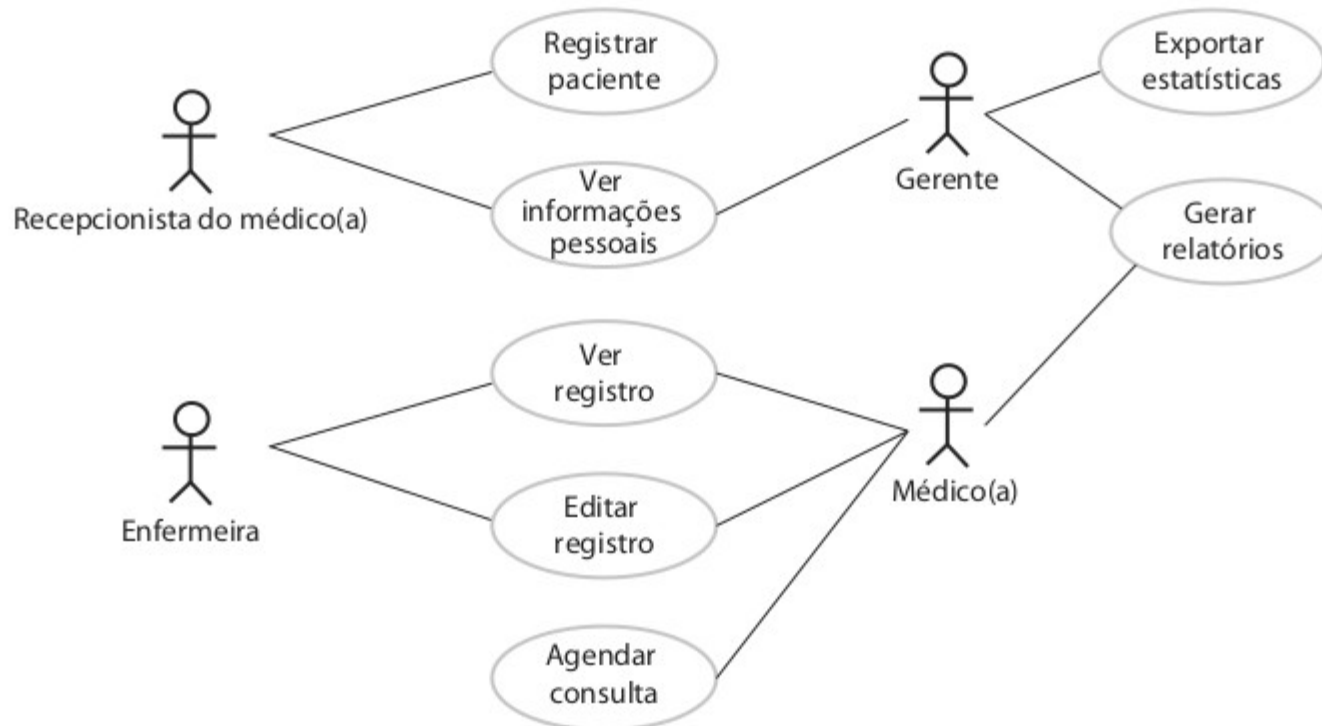
Atores, que podem ser pessoas ou outros sistemas, são representados como figuras 'palito'.

Cada classe de interação é representada por uma elipse. Linhas fazem a ligação entre os atores e a interação.

Opcionalmente, pontas de flechas podem ser adicionadas às linhas para mostrar como a interação se inicia.

**Figura 4.6**

Casos de uso para o MHC-PMS.



Não há distinção entre cenários e casos de uso que seja simples e rápida. Algumas pessoas consideram cada caso de uso um cenário único; outros, como sugerido por Stevens e Pooley (2006), encapsulam um conjunto de cenários em um único caso de uso.

Cada cenário é um segmento através do caso de uso. Portanto, seria um cenário para a interação normal além de cenários para cada possível exceção. Você pode, na prática, usá-los de qualquer forma.

# Dúvidas

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