Gerência de Requisitos.

Capítulo VII

Definir o Sistema

Aluizio Saiter, M. Sc.

Onde Estamos na Disciplina de Gerência de Requisitos?

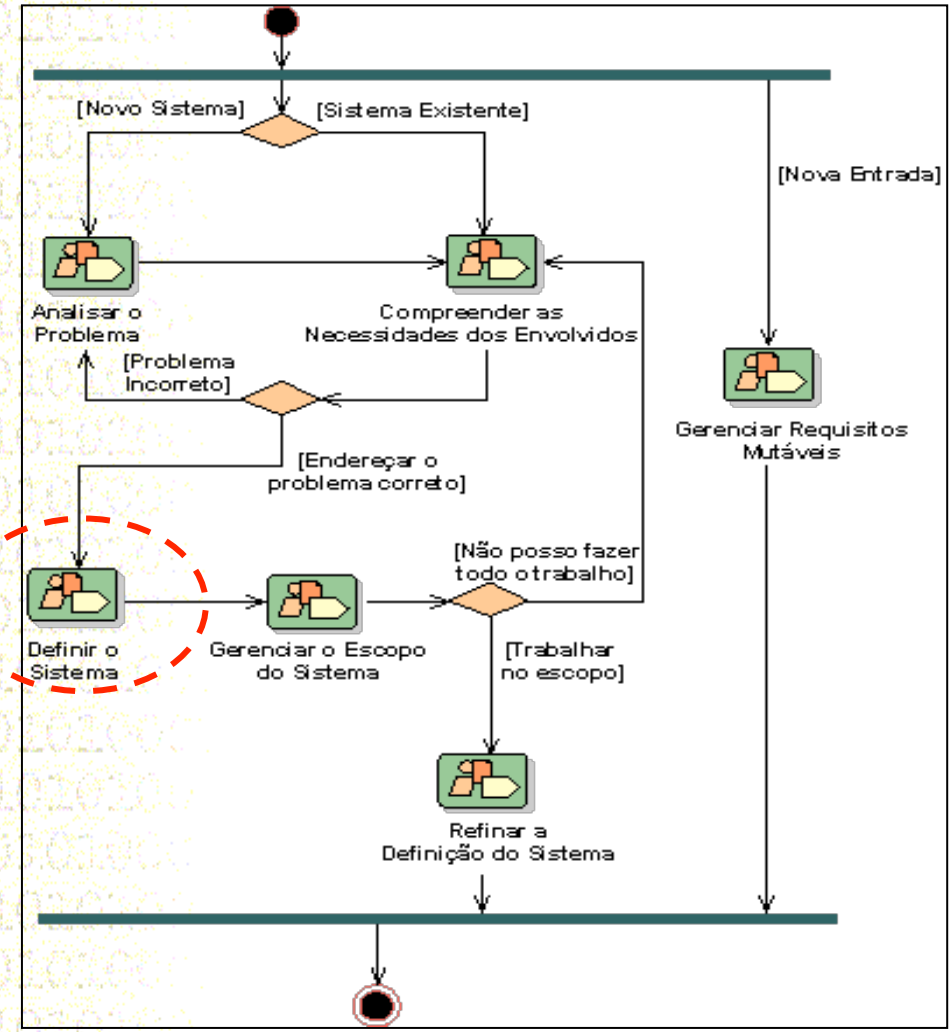
1. Sobre a disciplina de gerência de requisitos.
2. Boas práticas em engenharia de software.
3. Introdução a gerência de requisitos.
4. Introdução modelagem de casos de uso.
5. Analisar o problema.
6. Compreender as necessidades dos *stakeholders*.
7. **Definir o sistema.**
8. Gerenciar o escopo do sistema.
9. Refinar definição do sistema.
10. Controlar e gerenciar mudanças dos requisitos.
11. Estruturar os casos de uso.

Objetivos do Capítulo.

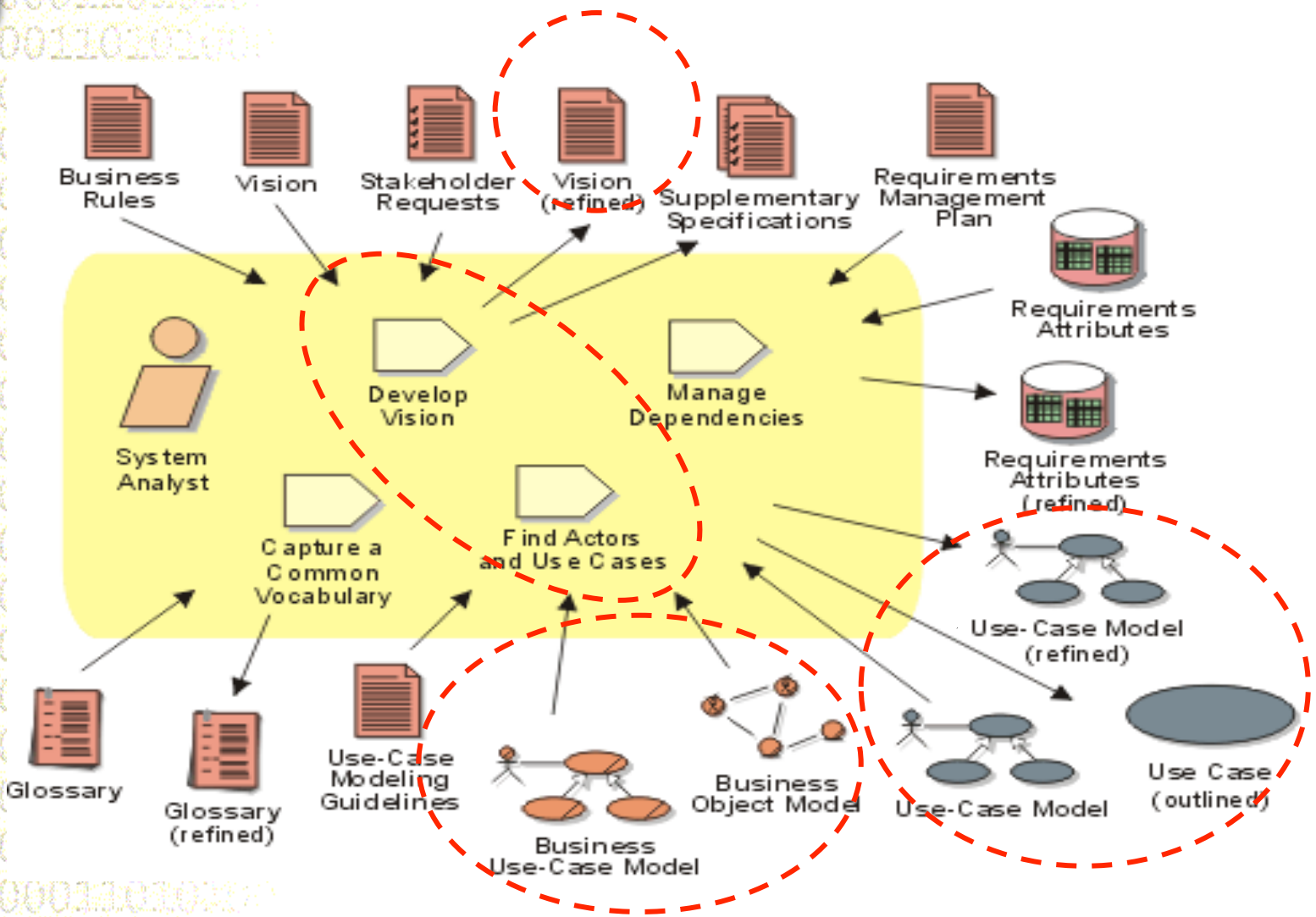
Este capítulo tem por objetivo apresentar ao aluno os seguintes conceitos:

- Refinar o documento de Visão
  - Definir o que são os features do produto.
  - Escrever o *product position statement*.
  - Identificar e documentar os features do produto a ser desenvolvido.
- Desenvolver o diagrama de casos de uso.
- Fazer o esboço da especificação dos casos de uso.
- Fluxo básico & Fluxos alternativos.
- Organizar o diagrama de casos de uso em packages.

Onde Estamos no *Workflow* da Disciplina de Requisitos no RUP©?



Definir o Sistema: Atividades e Artefatos.



O Domínio do Problema.

Antes de desenvolvermos o novo sistema nos devemos ter certeza de que entendemos as reais necessidades dos *stakeholders*. Estas necessidades residem no domínio do problema. Dean Leffingwell no artigo *Features, Use Cases, Requirements, Oh My!* define as necessidades dos *stakeholders* como:

> *". . .a reflection of the business, personal, or operational problem (or opportunity) that must be addressed to justify consideration, purchase, or use of a new system."*

No domínio do problema existem duas seqüências de passos, as quais devem ser executadas:

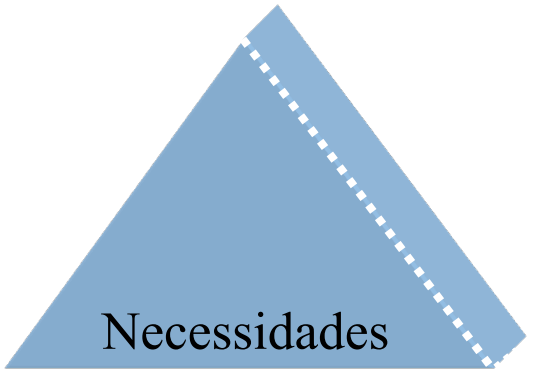
1. Análise do problema.
2. Entender as necessidades dos *stakeholders*.

Features, Use Cases, Requirements, Oh My!

O Domínio do Problema & as Necessidades dos *Stakeholders*.

As necessidades dos *stakeholders* são expressões relacionadas ao problema. Portanto, elas residem no domínio do problema. Elas não definem a solução, mas elas providenciam nossa primeira informação sobre qual deveria ser a solução.

Nos podemos visualizar a evolução da modelagem dos requisitos como uma pirâmide formada por vários níveis.



No nível mais alto reside o problema, e nos níveis mais baixos reside a solução.

Portanto, nos devemos colocar as informações relativas as necessidades dos *stakeholders* no topo da nossa pirâmide de requisitos.

O Domínio da Solução.

Felizmente as atividades executadas no domínio do problema não são de longa duração, complicadas e tão pouco demoradas. Uma vez finalizadas, nós iniciaremos as atividades relacionadas ao domínio da solução do problema.

Esta é a parte mais demorada e complicada da disciplina de requisitos: formular e formalizar uma solução para o problema.

No domínio da solução do problema existe uma série de atividades, as quais devem ser executadas com sucesso. Estas atividades incluem:

1. Definir o sistema.
2. Gerenciar o escopo e as solicitações de mudanças.
3. Refinar a solução do sistema.

O Domínio da Solução e o Conceito de *Feature*.

Quando nós pensamos em uma solução para o problema que foi identificado é natural que esta atividade tenha início pela descrição das características, capacidades e funcionalidades do sistema, ou seja, pelo seus features.

Os features tem um lugar importante na atividade de desenvolvimento de sistemas. Os *features* são expressões, as quais residem entre as necessidades reais dos usuários e a descrição de como estas necessidade serão solucionadas.

Podem existir várias soluções para as necessidades dos *stakeholders*. Portanto, a utilização do conceito de *features* prove também um limite no escopo da solução escolhida.

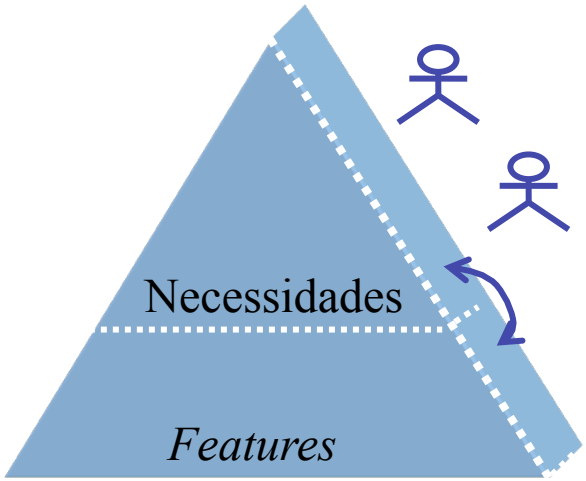
Dean Leffingwell no artigo *Features, Use Cases, Requirements, Oh My!* define *feature* como: *". . .a service that the system provides to fulfill one or more stakeholder needs."*

A Necessidade dos *Stakeholder* & os *Features*.

Os *features* são derivados das necessidades dos *stakeholders*. Portanto, nós os posicionaremos no segundo nível da pirâmide.

É importante ressaltar que os *features* não são apenas um refinamento (aumento do nível de detalhe) das necessidades dos *stakeholders*.

Na realidade, eles são uma resposta direta aos problemas descritos pelos stakeholders. Eles nos provêem uma descrição de alto nível da solução do problema.



Tipicamente nós devemos descrever o sistema por meio de 25-50 *features*, os quais caracterizam a ação do sistema. Se nós obtivermos mais do que 50 *features*, então nós temos um processo de abstração insuficiente, ou o sistema é muito complexo e deve ser dividido em partes.

O Conceito de *Feature* do Produto.

Um *feature* é uma solução ou serviço do sistema, o qual atende uma ou mais necessidade dos *stakeholders*. Ele é geralmente obtido de forma quase que imediata após a sessão de *brainstorming* com os *stakeholders* para capturar as necessidades dos mesmo.

Os *features* definem exatamente o que o sistema fará. Os *features* são descritos no documento de Visão. A diferença entre *feature* e casos de uso é que *features* são descritos com base na perspectiva do sistema, e os casos de uso são descritos com base na perspectiva do usuário.

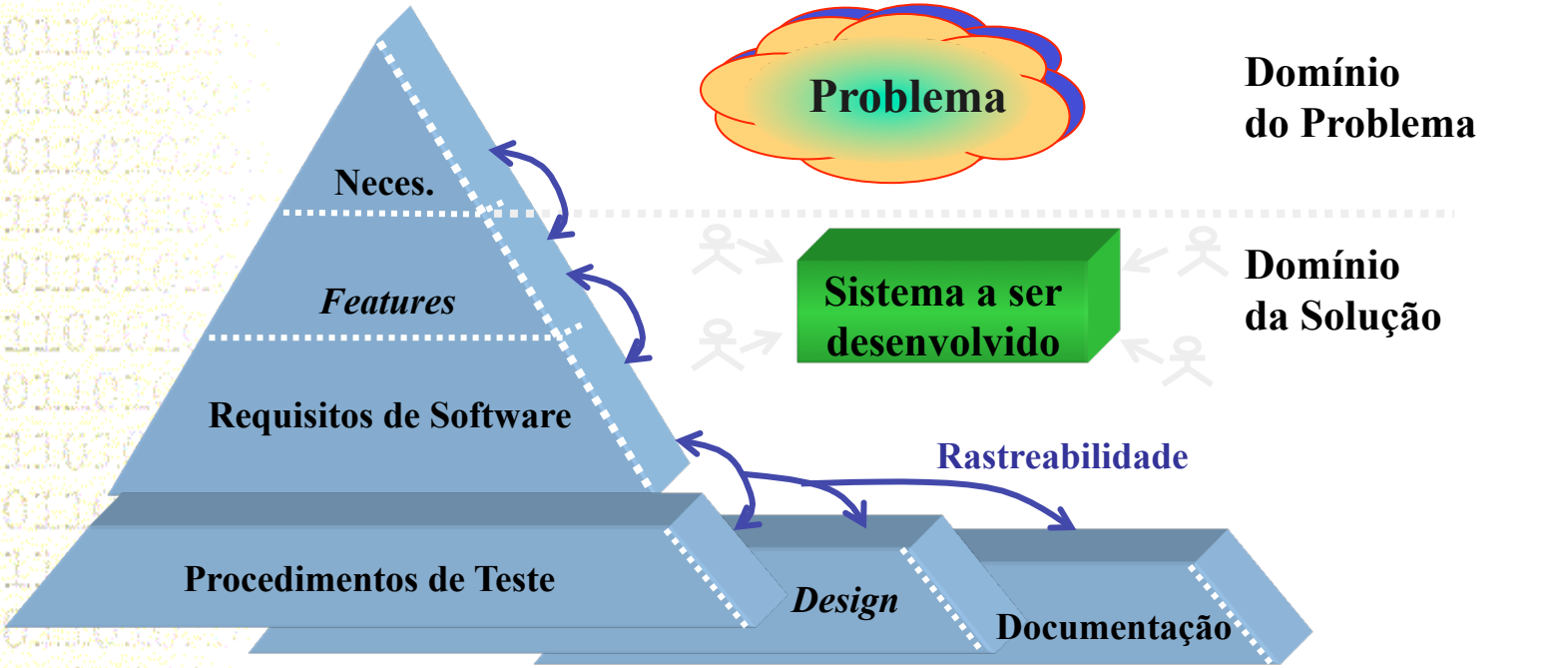
O documento de Visão é lido por várias pessoas no projeto. Ele é lido por pessoas com formação especifica em engenharia de software, e também por pessoas com formação mais voltada para a área de negócio.

Portanto, ele deve ter um formato geral. Os detalhes mais específicos são descritos nos casos de uso e na especificação suplementar.

Uma Visão da Hierarquia de Abstração dos Requisitos.

Modelar os requisitos envolve traduzir as necessidades e solicitações dos stakeholders em um conjunto de features de sistema.

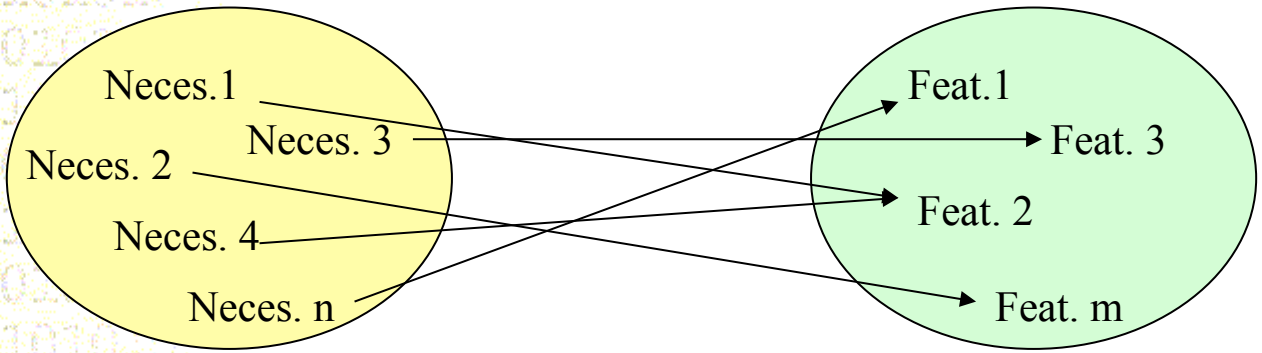
Estas por sua vez são detalhadas em especificações funcionais e não funcionais. As especificações são então detalhadas em design, procedimentos de teste, documentação do usuário etc.



O Relacionamento entre *Feature* & Necessidade dos *Stakeholders*.

O problema do negócio é identificado e documentado como necessidade dos *stakeholders*. As necessidades dos *stakeholders* são transformadas em *features* do produto.

Os *features* do produto e as necessidades dos *stakeholders* possuem relacionamento do tipo muitos-para-muitos, ou seja, um relacionamento do tipo (N x M). Algumas vezes um feature atende a uma ou mais necessidades dos *stakeholders*; outras vezes uma necessidade dos *stakeholders* é atendida por vários features do produto. O documento de Visão é a base dos features do produto.



Exemplo de *Feature* & Necessidade dos *Stakeholders*.

Cada *feature* corresponde a um serviço, o qual deve ser capaz de ser o seu resultado observado pelo usuário, operador ou sistema. O foco do *feature* deve ser na descrição da necessidade. A seguir é apresentado um exemplo de necessidade dos *stakeholder* e dos *features* associados.

- **Necessidade dos stakeholders:** Nos precisamos oferecer ao cliente condições do mesmo poder realizar operações de compra e venda ou consulta de ações em qualquer horário ou local.

- ***Feature 1:*** O sistema deverá ser implementado em browser para permitir o acesso dos cliente a partir de qualquer PC ligado a internet.

- ***Feature 2:*** O sistema deverá ser integrado com os sistemas XX, YY, ZZ e WW para permitir a realização de operações de consulta e comercialização de ações.

Qualidades dos Requisitos de Software - Revisão.

Ao identificar e documentar os *features* do sistema a ser desenvolvido é importante relembrarmos as características de qualidade dos requisitos.

.

- Correto.
- Completo.
- Consistente.
- Não ambíguo.
- Classificado por importância e estabilidade.
- Verificável.
- Modificável.
- Rastreável.
- Compreensível.

O *Product Position Statement.*

O *product position statement* sumariza a descrição da solução, a qual tem por objetivo o problema dos *stakeholders*. Ele enfatiza na importância da solução (produto). O *product position statement* tem por objetivo "vender" a idéia da solução utilizando poucas palavras. Ele deve ser descrito em uma página.

Para	(qual cliente)
Que	(declaração de necessidade)
O (nome do produto)	É um (categoria do produto)
Que	(declaração de benefícios)
Diferente	(alternativa competitiva primária)
Nosso produto	(declaração primária de diferenciação)

Exemplo de um *Product Position Statement.*

Para	**e-Commerce de Ações da Corretora Silva & Silva.**
Que	esta perdendo clientes a uma taxa de 100 clientes por mês.
e-Commerce StockBroker.	é um produto de comercialização de ações utilizando tecnologia Web.
Este sistema	possui todas as funcionalidade que os clientes desejam para facilmente poder comercializar ações.
Diferente de	continuar com a operação tradicional de atendimento de clientes por meio de telefonistas, ou comprar um produto de comercialização de ações, mas que não se ajusta totalmente as características da operação da Corretora Silva & Silva.
Nosso produto	proverá uma solução que atenderá todas as necessidades identificadas pelos *stakeholders* para atender os nossos clientes.

Exercício em Sala de Aula.

Exercício 7.1

Elaboração do *product positioning statement* e identificação dos *features* do sistema de e-Matrícula da Faculdade São José.

Exercício em Sala de Aula: Exercício 7.1.

1. Releia com atenção a especificação do sistema de e-Matrícula.

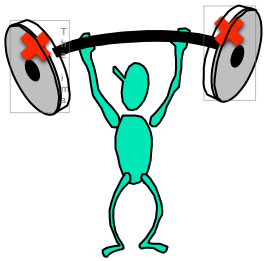
2. Descreva o *product position statement*. Sugestão: utilize o *problem statement* como ponto de partida.

3. Identifique possíveis *features* do sistema.

4. Atualize o documento de Visão.

5. Crie uma matriz de rastreabilidade relacionando *features* e necessidades dos *stakeholders*.

Tempo:
40 minutos.



O Artefato de Requisitos dos *Stakeholders*.

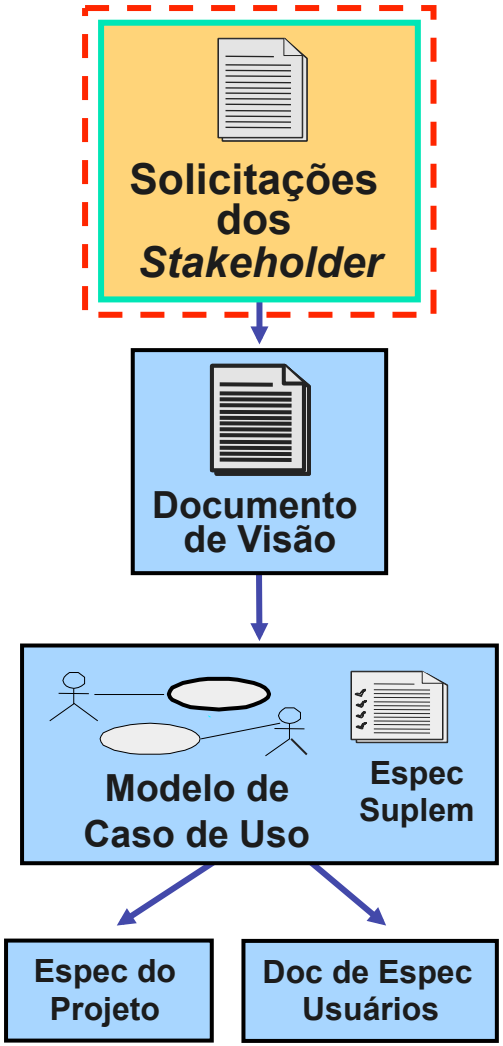
Necessidades
dos *Stakeholders*

↓

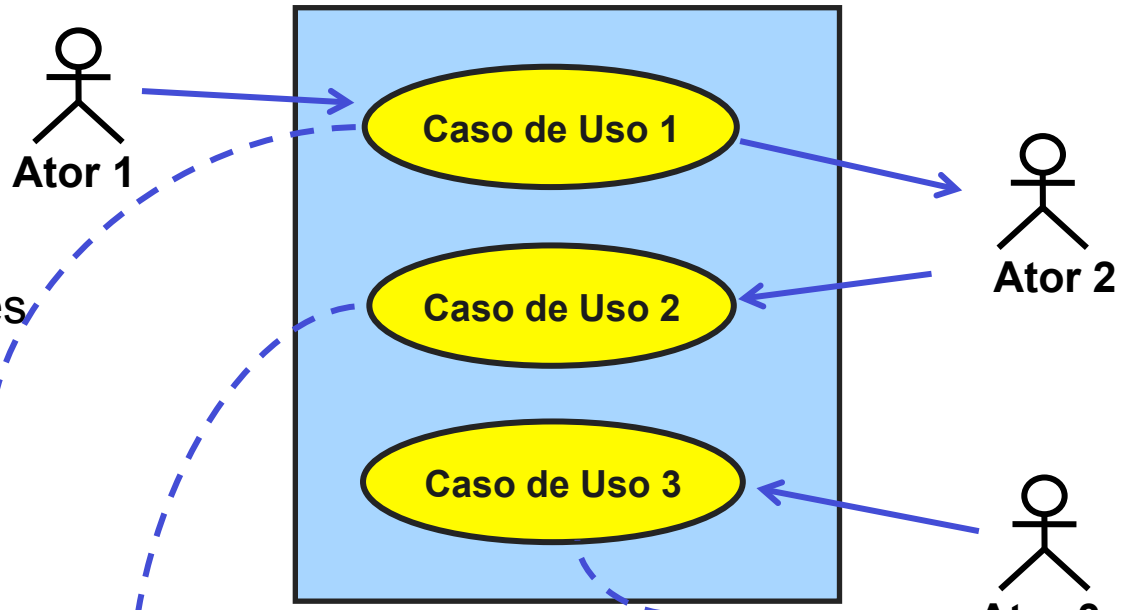
Características
e Necessidades

↓

**Requisitos
de Software**



Modelo de Caso de Uso - Revisão.



Modelo de Caso de Uso
- Comunicação entre Atores e Casos de Uso.
- Lista todos os Atores.
- Lista todos os Casos de Uso.

Caso de Uso 1 – Esp.
- Breve descrição
- Fluxo de eventos

Caso de Uso 2 – Esp.
- Breve descrição
- Fluxo de eventos

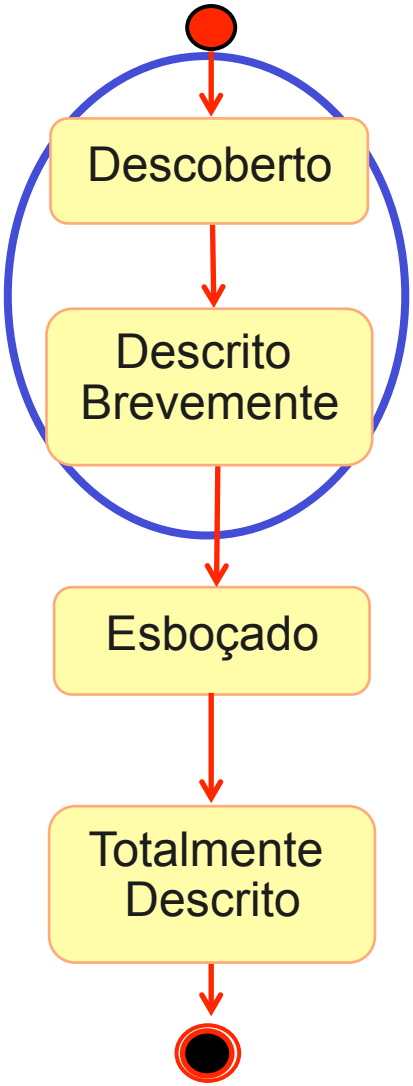
Caso de Uso 3 – Esp.
- Breve descrição
- Fluxo de eventos

Seqüência de Passos para Desenhar um Diagrama de Caso de Uso.

1. Identifique Atores e Casos de Uso.
   - Descrição breve.

2. Delineie cada caso de uso.
   - Fluxo básico.
   - Fluxos Alternativos.

3. Detalhe cada caso de uso.
   - Detalhes os fluxos de eventos.
   - Estruture cada fluxo de evento.
   - Adicione detalhes.
     - Pré e pós condições.
     - Requisitos especiais
     - etc.



Exercício em Sala de Aula.

Exercício 7.2

Identificação e descrição breve dos casos de uso do o sistema de e-Matrícula da Faculdade São José.

Exercício em Sala de Aula: Exercício 7.2.

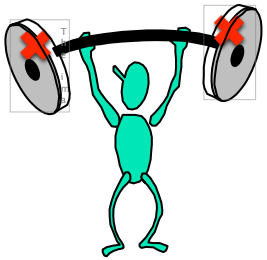
1. Releia com atenção a especificação do SoW do sistema de e-Matrícula.

2. Reveja os atores e os casos de uso identificados anteriormente, e verifique se existem novos atores e casos de uso.

3. Documente os casos de uso e os atores. Sugestão: veja o exemplo do sistema de *e-Commerce* de Ações da Corretora Silva e Silva.

4. Atualize os documentos: diagrama de caso de uso e especificação de caso de uso.

Tempo:
30 minutos.



Seqüência de Passos para Desenhar um Diagrama de Caso de Uso.

Identifique os atores e casos de uso.

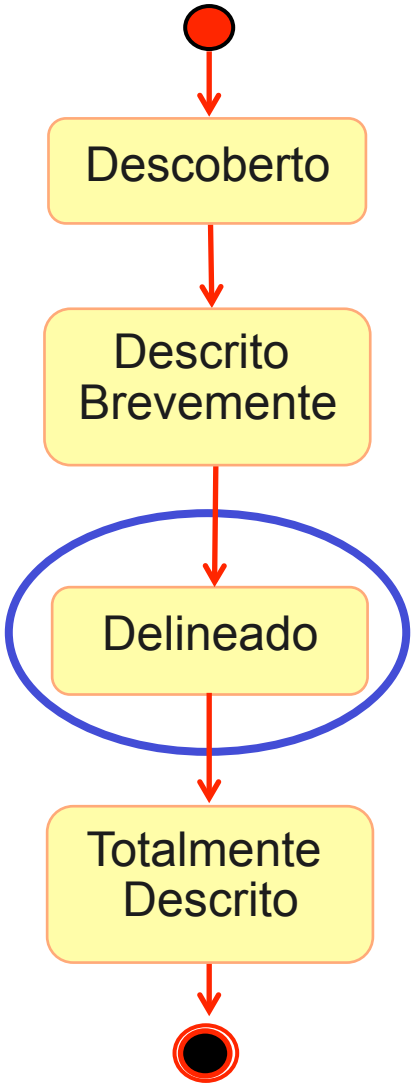
- Descrição breve.

Delineie cada caso de uso.

- **Fluxo básico.**
- **Fluxos alternativos.**

Detalhe cada caso de uso.

- Detalhes os fluxos de eventos.
- Estruture cada fluxo de evento.
- Adicione detalhes.
  - Pré e pós condições.
  - Requisitos especiais
  - etc.



Esboce O Fluxo de Cada Caso de Uso.

Desenvolver a seqüência de ações dos casos de uso é um processo iterativo. Ele tem início com o desenvolvimento de um draft da seqüência de ações, tal como no exemplo abaixo. O passo seguinte é então adicionar a descrição dos fluxos. O draft não é a versão final da especificação do caso de uso, mas sim o seu ponto de partida.

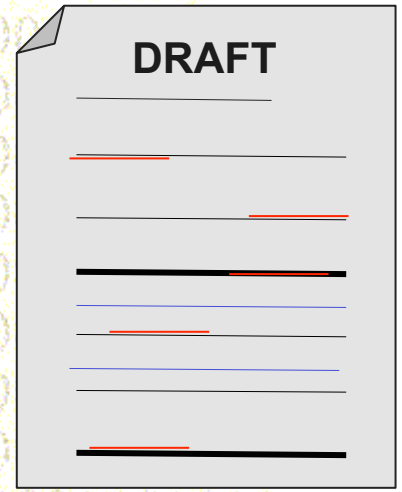
Nome do caso de uso
Breve descrição
Fluxo básico
1. Primeiro passo
2. Segundo passo
3. Terceiro passo

A1 Fluxo alternativo 1
A2 Fluxo alternativo 2
A3 Fluxo alternativo 3

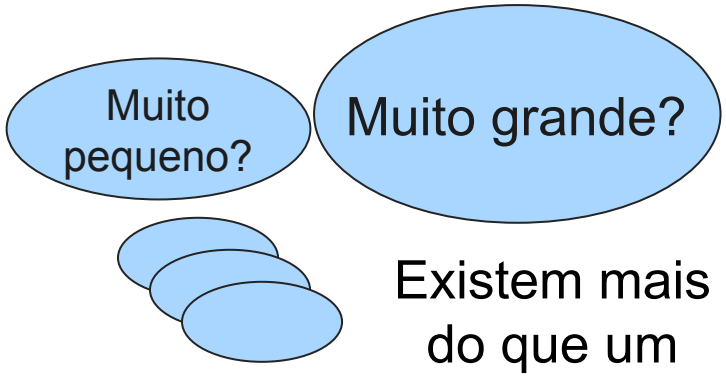
Numere ou marque os passos.

Estrutura do fluxo em passos.

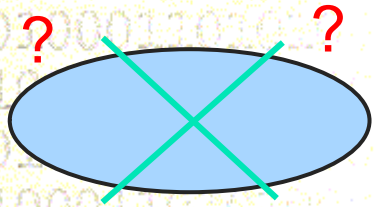
Porque Fazer o Esboço dos Casos de Uso?



Tamanho dos casos de uso



Existem mais do que um caso de uso?



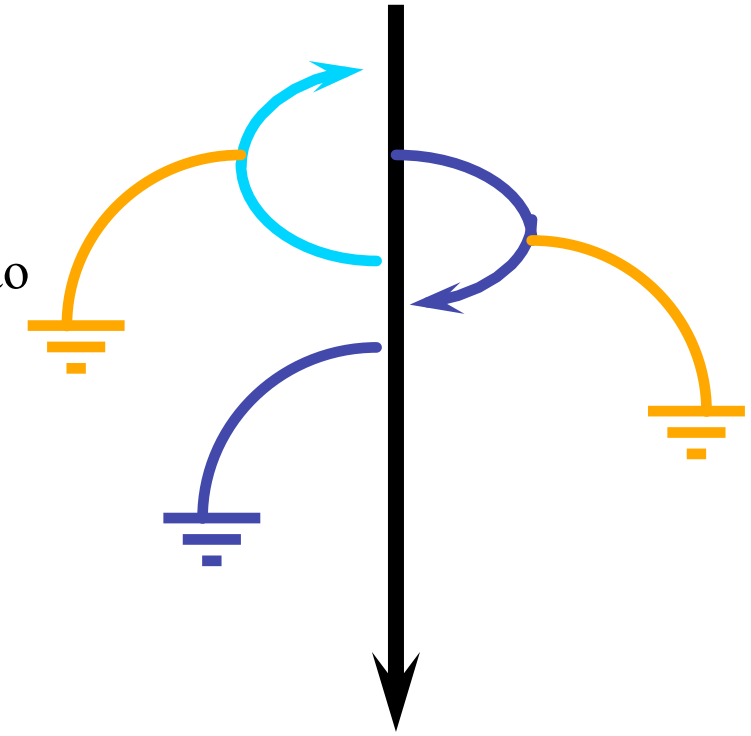
caso de uso ?

Esboçar os casos de uso ajuda a identificar os fluxos

Os Fluxo de Eventos Básico & Alternativos.

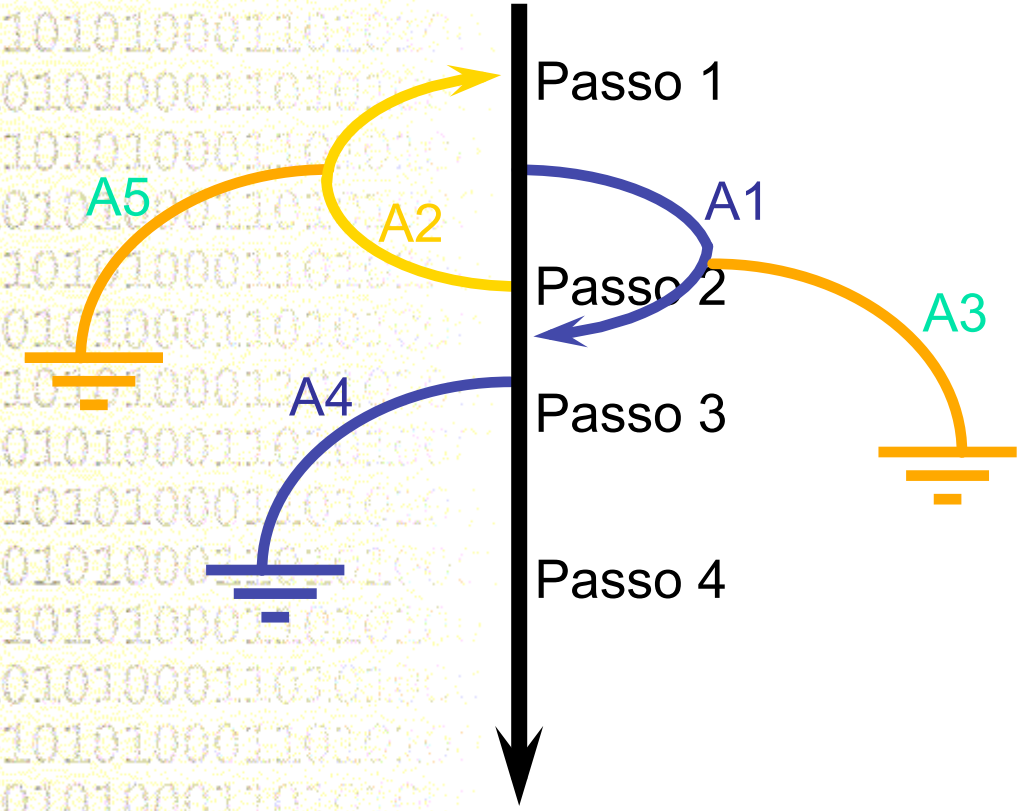
O draft do caso de uso, ou seja o caso de uso delineado, é utilizado como base para a descrição completa da especificação do caso de uso. A estrutura do fluxo de eventos deve ser feita de forma que seja fácil de ser seguida em qualquer cenário, e ser capaz de entender o que ocorre em cada fluxo alternativo.

- Um fluxo básico somente.
  - Cenário do tipo *happy day*.
  - Execução do cenário do início ao final.
- Vários fluxos alternativos.
  - Variações do cenário básico.
  - Situações diferentes.
  - Erros e exceções.



Representação dos Fluxos Básicos e Alternativos.

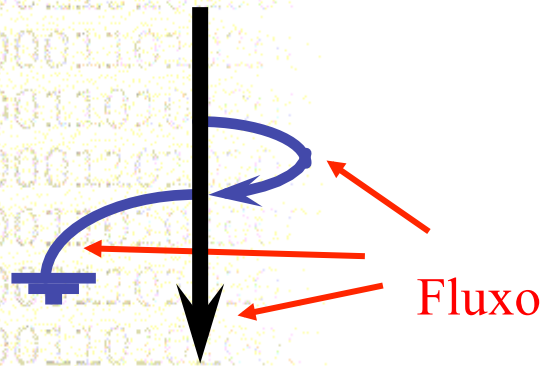
Os fluxos básicos e alternativos são descritos em suas respectivas seções no documento de especificação do caso de uso. Cada fluxo tem a sua respectiva seção.

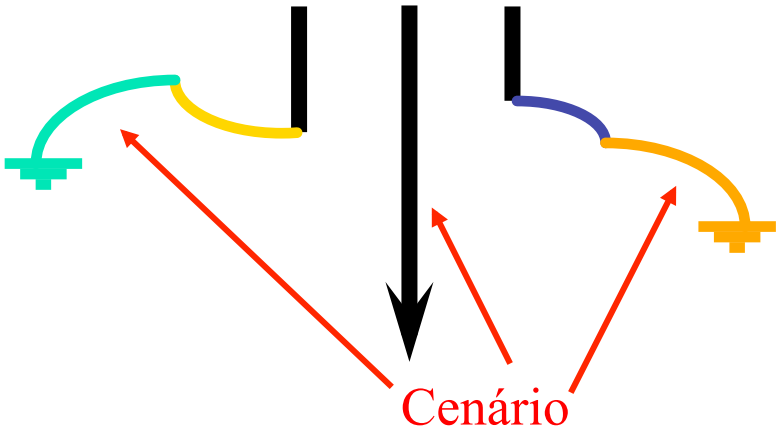


<Nome do Caso de Uso>
1. Breve descrição
2. Fluxo de Eventos
   2.1 Fluxo básico
      Passo 1
      Passo 2
      Passo 3
      Passo 4
   2.2 Fluxos Alternativos
      2.2.1 A1 …
      2.2.2 A2 …
      2.2.3 A3 …
      2.2.4 A4 …
      2.2.5 A5 …

O Conceito de Cenário do Caso de Uso.

O cenário descreve uma instância do caso de uso, ou seja o caso de uso executado por determinado conjunto de valores e eventos. Ele representa uma caminho do ponto inicial ate um ponto final do fluxo de eventos. O cenário do caso de uso pode envolver o fluxo básico somente, ou o fluxo básico e os fluxos alternativos em qualquer combinação. Contudo, sempre começando pelo fluxo básico.





Fluxo: uma seqüência de passos.

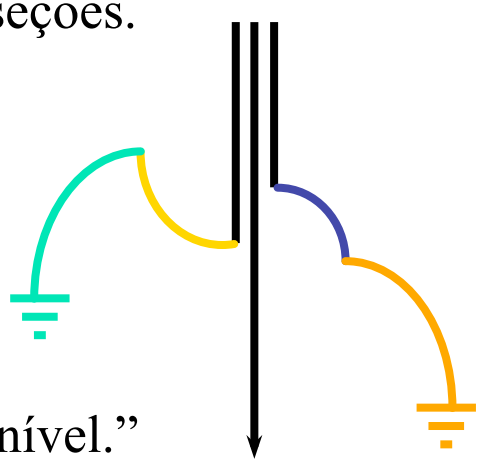
Caso de Uso: Um *conteiner* que descreve os fluxos.

Cenário: Uma seqüência de passos do início ao fim do caso de uso.

A Captura dos Cenários dos Casos de Uso.

Cada cenário captura uma família de casos de testes. Cada família de caso de teste segue um determinado caminho do fluxo de eventos do caso de uso, mas utiliza diferentes valores para assegurar as possíveis combinações do teste sejam garantidas.

- Descreva os cenários dos casos de uso em suas seções.
- Dê a cada cenário um nome.
- Liste o nome de cada fluxo no cenário.
- Coloque os fluxos em seqüências.



Exemplo: Caso de Uso Obter Cotação.

- Cenário “Comunicação com o *Host* esta indisponível.”
- Fluxos “Fluxo básico” + “Sistema de cotação esta indisponível.”

O Esboço dos Fluxos de Eventos.

Abaixo nos apresentamos uma lista de decisões para ajudar na identificação de fluxos de eventos alternativos. Fluxos alternativos descrevem como o sistema deve proceder quando algo não funciona como o esperado.

Quando trabalhamos com fluxos alternativos nos podemos utilizar de algumas perguntas para nos ajudar a identificá-los. Estas questões estão relacionadas ao funcionamento do sistema e como eles esperam interagir com o sistema quando algo inusitado acontece.

- Existem situações adicionais no caso de uso?
- Quais são as diferenças?
- Quais são as variações?
- O que pode ir errado?
- O que pode não acontecer?
- Que tipos de recursos podem ser bloqueados?

Esboço da Seqüência de Passos: Obter Cotação.

Fluxo básico.

1. Cliente faz *log in*.
2. Cliente seleciona opção de obter cotação.
3. Cliente informa símbolo da ação.
4. Obtém cotação do Sistema de Cotação.
5. Apresenta cotação.
6. Cliente seleciona outras cotações.
7. Cliente faz *logs off*.

Fluxos alternativos

A1. Cliente não foi identificado.
A2. Sistema de Cotação esta indisponível.
A3. Símbolo da empresa não existe.

Quais são as alternativas?

Exercício em Sala de Aula.

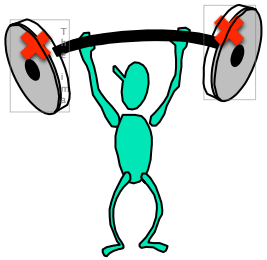
Exercício 7.3

Identificação e descrição breve dos casos de uso do o sistema de e-Matrícula da Faculdade São José.

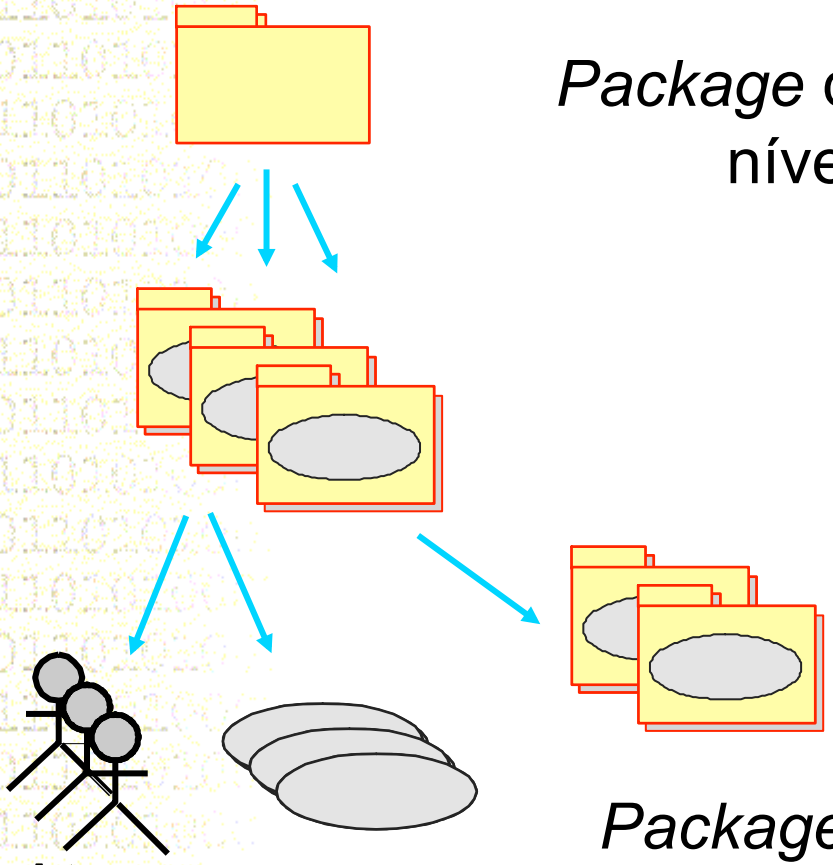
Exercício em Sala de Aula: Exercício 7.3.

1. Releia com atenção a descrição do SoW do sistema de e-Matrícula.
2. Selecione um dos casos de uso identificados no exercício 7.2.
3. Escreva a seqüência de passos.
   - Fluxo básico.
   - Fluxo alternativo.
4. Nomeie e documente os cenários.

Tempo:
40 minutos.



Packages: A Organização do Modelo de Casos de Uso.



Questões de Revisão.

1. O que é *product position statement*?

2. Quais são algumas das perguntas que podem serem feitas para identificar os Casos de Uso?

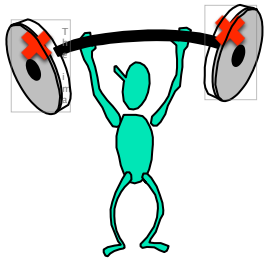
3. Porque documentar os cenários?

4. Qual é a diferença entre Caso de Uso, fluxo e cenário?

5. O que esta incluído no *outline* do Caso de Uso?

6. Como pode ser organizado o modelo de Caso de Uso?

Data de Entrega:
16 de Abril



That's All Folks!