

Análise e Programação Orientada a Objetos

Sistemas de Informação

Unidade II – Parte II
Análise de Requisitos
Casos de Uso

Prof. Marciel de Liz Santos

Relembrando...

- ◆ Orientação a Objetos
 - Classes e objetos
 - Mensagens
 - Abstração
 - Encapsulamento
 - Polimorfismo
 - Herança
- ◆ UML - linguagem padrão para modelar sistemas orientados a objetos.

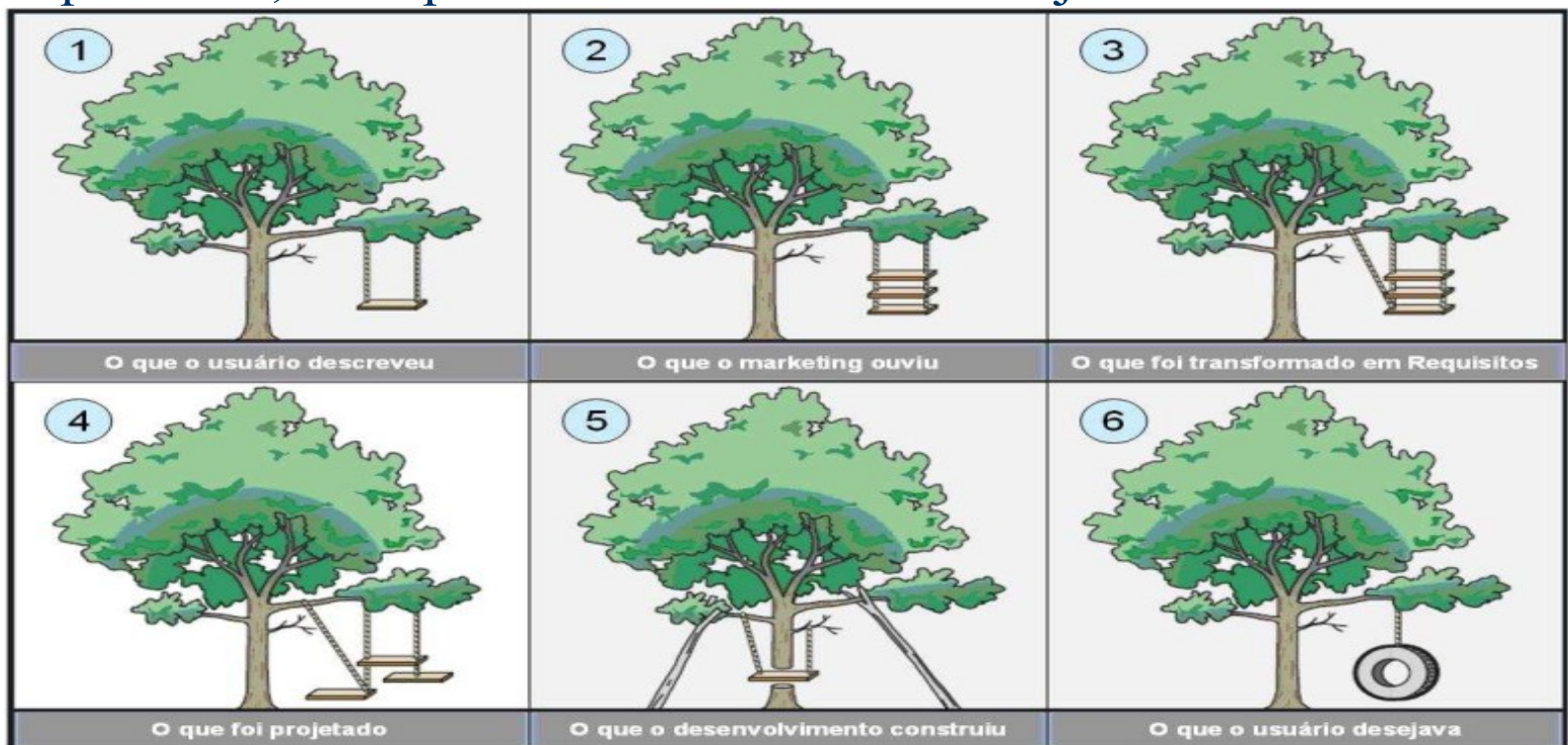


Análise de Requisitos

Levantamento de Requisitos

O que o cliente desejava...

- ♦ Uma das maiores fontes de fracasso é desenvolver softwares de qualidade, mas que não satisfazem aos desejos dos clientes



Levantamento de Requisitos

- ◆ Requisitos
 - Conjunto de necessidades estabelecido pelo cliente/usuário do sistema que definem a estrutura e o comportamento do software que será desenvolvido.
- ◆ Deve prever o que o software deve fazer sem a preocupação de como será construído.
- ◆ Projeto bem sucedido
 - usuário participa mais do processo de desenvolvimento
 - É o usuário quem vai medir o nível de qualidade do software
- ◆ Independente do método que será utilizado

Levantamento de Requisitos (Técnicas)

- ◆ Entrevistas
- ◆ Análise de Documentos
- ◆ Observação Direta
- ◆ Seções Brainstorming
- ◆ Reuso de requisitos

Artefatos Produzidos

- ◆ Sumário Executivo/Visão Geral
- ◆ Documento de Requisitos
(Funcionais e Não-Funcionais)
- ◆ Glossário

Sumário Executivo

- ◆ Também chamado de Visão Geral do Sistema
- ◆ Deve descrever aquilo que conseguiu-se descobrir de relevante sobre o sistema após as conversas com os clientes e usuários.
- ◆ 2 páginas, no máximo

Sumário Executivo / Visão Geral do Sistema

- ♦ documento de texto em formato livre

Sistema Videolocadora

Visão Geral do Sistema

É proposto o desenvolvimento de um sistema de controle de videolocadora, que vai informatizar as funções de empréstimo, devolução e reserva de fitas. O objetivo do sistema é agilizar o processo de empréstimo e garantir maior segurança, ao mesmo tempo que possibilita um melhor controle das informações por parte da gerência. Deverão ser gerados relatórios de empréstimos por cliente, empréstimos por fita e empréstimos no mês. O sistema deverá calcular automaticamente o valor dos pagamentos a serem efetuados em cada empréstimo inclusive multas e descontos devidos. A cada devolução de fitas corresponderá um pagamento, não sendo possível trabalhar com sistema de créditos. A impossibilidade de efetuar um pagamento deve deixar o cliente suspenso, ou seja, impossibilitado de emprestar novas fitas até saldar a dívida.

Documento de Requisitos

- ♦ Requisitos **funcionais** indicam as funções que são executadas pelo sistema
 - **evidentes** são efetuados com conhecimento do usuário. Ocorrem pela interface do sistema
 - **ocultos** são efetuados pelo sistema sem o conhecimento explícito do usuário
- ♦ Requisitos **não funcionais** são restrições que se coloca sobre como o sistema deve realizar seus requisitos funcionais
 - **obrigatórios / desejados**
 - **permanente / transitório**

Requisitos Não-Funcionais

- ♦ Categoria:
 - de interface
 - de especificação
 - de eficiência
 - de segurança
 - ...
- ♦ Associação
 - Associados a requisitos funcionais
 - Suplementares

Tabela de Requisitos Funcionais

- ◆ Código do requisito funcional (Ex.: F1, F2, F3, ...).
- ◆ Nome do requisito funcional (especificação curta).
- ◆ Descrição (especificação longa e detalhamento do requisito).
- ◆ Categoria funcional: evidente ou oculto.

Tabela de Requisitos Não-Funcionais

- ◆ Código do requisito não funcional (Ex.: NF1.1, NF1.2, ... NF2.1, NF2.2, ...).
- ◆ Nome do requisito não funcional (especificação curta).
- ◆ Restrição: especificação (longa) do requisito não funcional.
- ◆ Categoria: tipo de restrição: segurança, performance, compatibilidade, etc.
- ◆ Obrigatoriedade: se o requisito é desejável ou obrigatório.
- ◆ Permanência: se o requisito é permanente ou transitório.

Requisitos Funcionais e Não-Funcionais Associados

F1 Registrar empréstimos		Oculto ()		
Descrição: O sistema deve registrar empréstimos de fitas, indicando o cliente e as fitas que foram emprestadas, bem como a data do empréstimo e valor previsto para pagamento na devolução.				
Requisitos Não Funcionais				
Nome	Restrição	Categoria	Desejável	Permanente
NF1.1 Controle de Acesso	A função só pode ser acessada por usuário com perfil de operador ou superior.	Segurança	()	(x)
NF1.2 Identificação de Fitas	As fitas devem ser identificadas por um código de barras	Interface	()	(x)
NF1.3 Identificação do cliente	O cliente deverá ser identificado a partir de seu nome	Interface	()	()
NF1.4 Tempo de registro	O tempo para registro de cada fita deve ser inferior a um segundo.	Desempenho	(x)	()
NF1.5 Janela única	Todas as funções relacionadas a empréstimos devem ser efetuadas em uma única janela	Interface	(x)	(x)
...

F2 Calcular descontos		Oculto (x)		
Descrição: O sistema deve calcular descontos nos empréstimos em função da política da empresa.				
Requisitos Não Funcionais				
Nome	Restrição	Categoria	Desejável	Permanente
NF2.1 Desconto de fim de semana	Nos fins de semana, usuários que levam 4 fitas pagam apenas 3.	Especificação	()	()
...

Requisitos Suplementares

Nome	Restrição	Categoria	Desejável	Permanente
S1 Tipo de Interface	As interfaces do sistema devem ser implementadas como formulários acessíveis em um browser html.	Interface	()	()
S2 Armazenamento de dados	A camada de persistência deve ser implementada de forma que diferentes tecnologias de bancos de dados possam vir a ser utilizadas no futuro	Persistência	()	(x)
S3 Perfis de usuário	Os perfis de usuário para acesso ao sistema são: 3. Administrador - pode efetuar todas as operações. 2. Operador - pode efetuar as operações de empréstimo, devolução, pagamento e cadastramento. 1. Convidado - pode efetuar apenas consultas nos próprios dados (cliente).	Segurança	()	()
...

Glossário

- ◆ Definir termos
- ◆ Evita ambiguidades
 - No exemplo da videolocadora
 - Um cliente aluga Filmes ou Fita?
- ◆ Exemplo (hotel):
 - **Estadia:** Período em que um aposento é alugado a uma pessoa, compreendido do meio dia da entrada deste, até o meio dia do dia subsequente.



Modelagem de Casos de Uso

Introdução

- ♦ **Casos de uso** = processos de negócio da empresa
- ♦ O **modelo de casos de uso (MCU)** é uma representação das funcionalidades externamente observáveis do sistema e dos elementos externos ao sistema
- ♦ O MCU modela os **requisitos funcionais** do sistema
- ♦ Um caso de uso pode estar associado a **um** ou **mais** requisitos funcionais
- ♦ O diagrama da UML utilizado na modelagem de casos de uso é o **diagrama de casos de uso**.

Introdução

- ◆ Técnica de modelagem idealizada por Ivar Jacobson, na década de 1970
- ◆ Em 1992 foi incorporada ao processo de desenvolvimento Objectory
- ◆ Posteriormente, a notação foi adicionada à UML.
- ◆ Modelo direciona diversas das tarefas posteriores do ciclo de vida do sistema de software
- ◆ Força os desenvolvedores a **moldar o sistema de acordo com as necessidades do usuário**
- ◆ Usa **notação gráfica** e descrição em **linguagem natural**

Casos de Uso

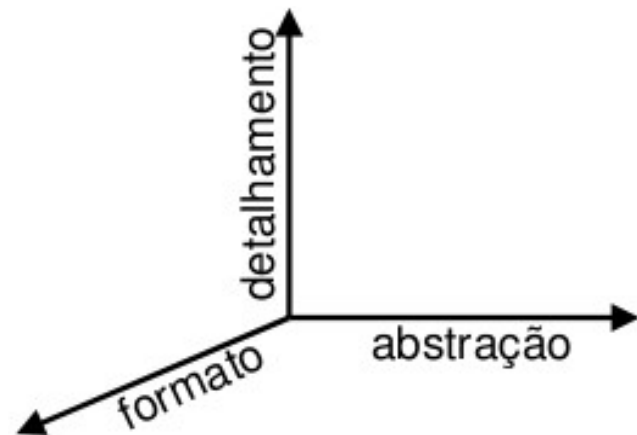
- ♦ Um caso de uso é a especificação de uma **seqüência completa de interações** entre um sistema e os agentes externos
- ♦ São usados para capturar comportamentos relevantes do sistema, sem a preocupação de especificar como serão implementados
- ♦ Um **modelo de casos** de uso típico é formado de vários casos de uso
- ♦ Composto de:
 - Casos de uso
 - Atores
 - Relacionamentos entre eles

Casos de Uso

Um caso de uso representa quem faz o que (interage) com o sistema, sem considerar o comportamento interno do sistema.

Casos de Uso – Descrições narrativas

- ◆ Cada caso de uso é definido através da descrição narrativa das interações que ocorrem entre o(s) elemento(s) externo(s) e o sistema.
- ◆ A UML não define uma estrutura textual
- ◆ Há vários estilos, a escolha fica a cargo da equipe
- ◆ 3 dimensões para se descrever casos de uso.
 - Formato (estrutura)
 - Grau de detalhamento
 - Grau de abstração
- ◆ Formato:
 - Descrição contínua
 - Descrição numerada
 - Narrativa particionada



Exemplo de descrição contínua

◆ Caso de uso **Realizar Saque**

O Cliente chega ao caixa eletrônico e insere seu cartão. O Sistema requisita a senha do Cliente. Após o Cliente fornecer sua senha e esta ser validada, o Sistema exibe as opções de operações possíveis. O Cliente opta por realizar um saque. Então o Sistema requisita o total a ser sacado. O Sistema fornece a quantia desejada e imprime o recibo para o Cliente

Exemplo de descrição numerada

1. Cliente insere seu cartão no caixa eletrônico.
2. Sistema apresenta solicitação de senha.
3. Cliente digita senha.
4. Sistema exibe menu de operações disponíveis.
5. Cliente indica que deseja realizar um saque.
6. Sistema requisita quantia a ser sacada.
7. Cliente retira a quantia e recibo.

Exemplo de descrição narrativa particionada

Cliente	Sistema
Insere seu cartão no caixa eletrônico.	Apresenta solicitação de senha.
Digita senha.	Exibe operações disponíveis.
Solicita realização de saque.	Requisita quantia a ser sacada.
Retira a quantia e o recibo.	

Detalhamento

- ♦ O grau de detalhamento a ser utilizado na descrição de um caso de uso também pode variar.
- ♦ Um caso de uso sucinto descreve as interações sem muitos detalhes.
- ♦ Um caso de uso expandido descreve as interações em detalhes.

Grau de Abstração

- ♦ Diz respeito à existência ou não de **menção à tecnologia** a ser utilizada na descrição deste caso de uso.
- ♦ Um caso de uso **essencial** não faz menção à tecnologia a ser utilizada.
- ♦ Um caso de uso **real** apresenta detalhes da tecnologia a ser utilizada na implementação deste caso de uso
 - Ex: caixa eletrônico

Cenários

- ♦ A funcionalidade descrita por um caso de uso tem diversas maneiras de ser utilizada.
- ♦ Um **cenário** é a descrição de uma das maneiras pelas quais um caso de uso pode ser utilizado.
- ♦ Um cenário também é chamado de instância de um caso de uso.
- ♦ Normalmente há diversos cenários para um mesmo um caso de uso.
- ♦ Úteis durante a modelagem de interações.
- ♦ Caso de Uso x Cenário

Exemplo de Cenários (Compra Internet)

- ♦ O Cliente seleciona um conjunto de produtos do catálogo da loja.
- ♦ Após selecionar os produtos, o cliente indica o desejo de realizar o pagamento por cartão de crédito.
- ♦ O sistema informa que o último produto não tem em estoque.
- ♦ O cliente pede para que o sistema feche a compra sem o aquele produto
- ♦ O sistema solicita os dados do cartão, data da expiração, e o endereço para entrega dos produtos.
- ♦ O sistema apresenta o valor total, a data de entrega e uma identificação do pedido para futuro rastreamento.
- ♦ O sistema também envia para o cliente, via correio eletrônico, uma confirmação do pedido de compra.
- ♦ O sistema envia os dados do pedido para o sistema de logística da empresa.

Cenário

- ♦ Uma coleção de cenários pode ser utilizada
- ♦ posteriormente na fase de testes para verificar a existência de erros na implementação
- ♦ Durante a construção de um cenário podem ser identificados
 - novos detalhes do caso de uso
 - novos casos de uso
- ♦ No exemplo anterior:
 - O que acontece se o cliente sair do sistema antes de concluir a compra?
 - E se o cartão não for aceito?
 - O sistema de logística é um ator do sistema?

Atores

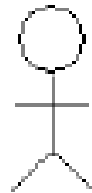
- ◆ Elemento externo que interage DIRETAMENTE com o sistema.
 - “externo”: atores não fazem parte do sistema.
 - “interage”: um ator troca informações com o sistema.
- ◆ Casos de uso representam uma seqüência de interações entre o sistema e o ator.
- ◆ Normalmente um ator inicia a seqüência de interações com o sistema ou um evento acontece para que o sistema responda.

Atores

♦ Categorias de atores:

- **peessoas** (Empregado, Cliente, Gerente, Almoxarife, Vendedor, etc);
- **organizações** (Empresa Fornecedora, Agência de Impostos, Administradora de Cartões, etc);
- **outros sistemas** (Sistema de Cobrança, Sistema de Estoque de Produtos, etc).
- **equipamentos** (Leitora de Código de Barras, Sensor, etc.)

♦ Em UML



Nome do Ator

Atores

- ♦ Um ator corresponde a um **papel** representado em relação ao sistema.
 - O **mesmo indivíduo** pode ser o **Cliente** que compra mercadorias e o **Vendedor** que processa vendas.
 - **Uma pessoa** pode representar o papel de **Funcionário** de uma instituição bancária que realiza a manutenção de um caixa eletrônico, mas também pode ser o **Cliente** do banco que realiza o saque de uma quantia.
- ♦ O nome dado a um ator deve lembrar **o seu papel**, ao invés de lembrar **quem o representa**.

Atores

- ♦ Um ator pode participar de muitos casos de uso
- ♦ Um caso de uso pode envolver vários atores
 - **Primário:** é aquele que inicia uma seqüência de interações de um caso de uso.
 - **Secundários:** supervisionam, operam, mantêm ou auxiliam na utilização do sistema.
- ♦ Exemplo: para que o Usuário (primário) requisiite uma página a um Browser (sistema), um outro ator (secundário) está envolvido, o Servidor Web.

Relacionamentos

- ◆ Casos de uso e atores não existem sozinhos. Deve haver relacionamentos entre eles
- ◆ A UML define diversos tipos de relacionamentos no modelo de casos de uso:
 - Comunicação
 - Inclusão
 - Extensão
 - Generalização

Notação (Exemplo)

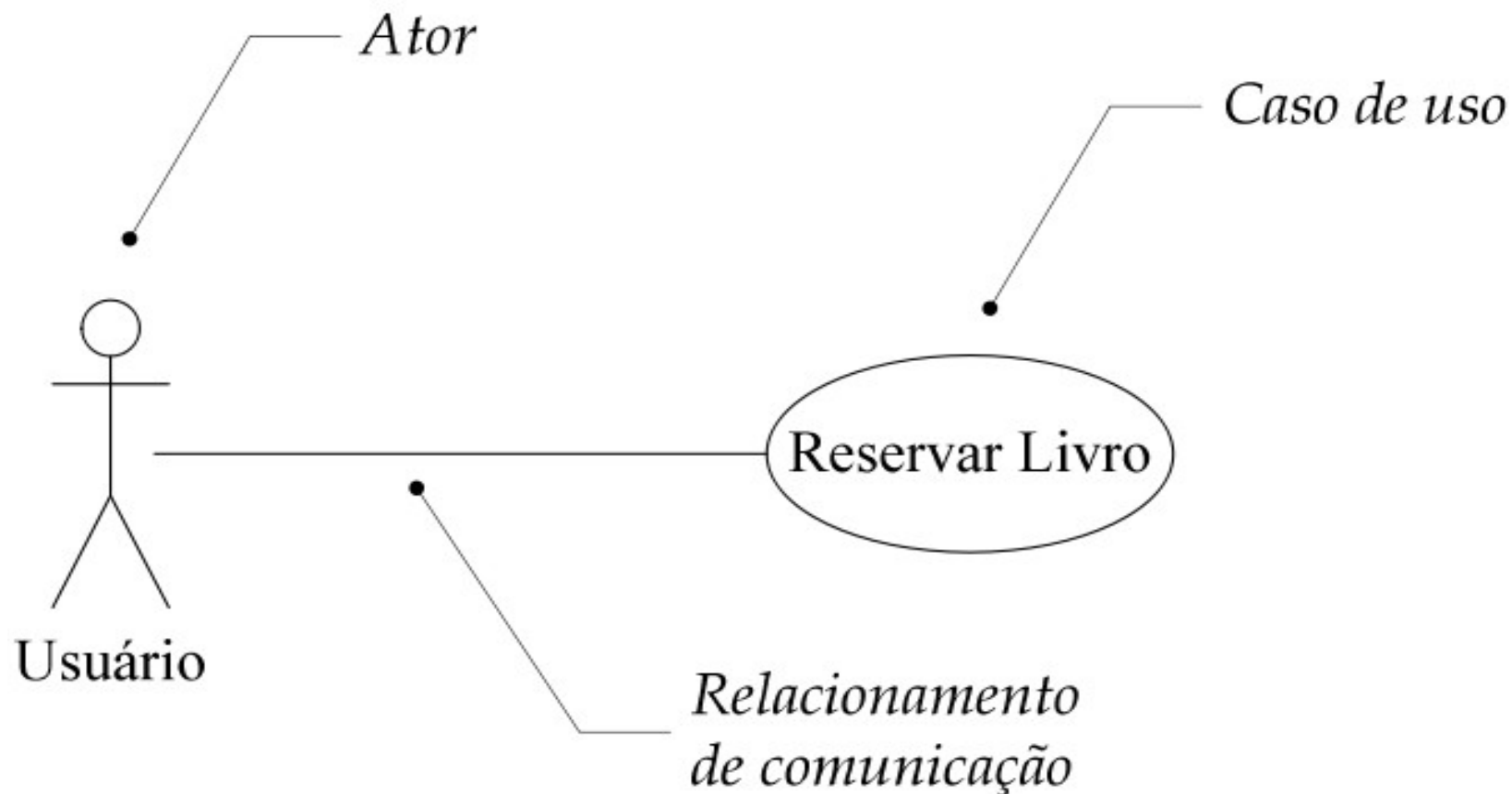
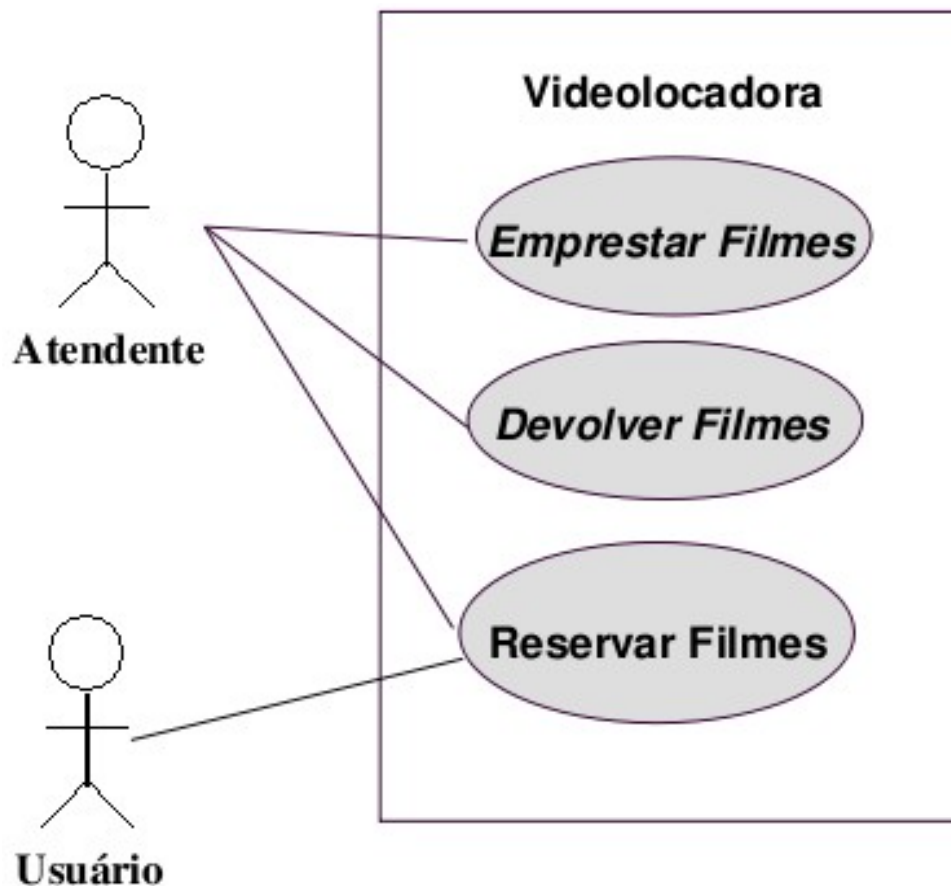
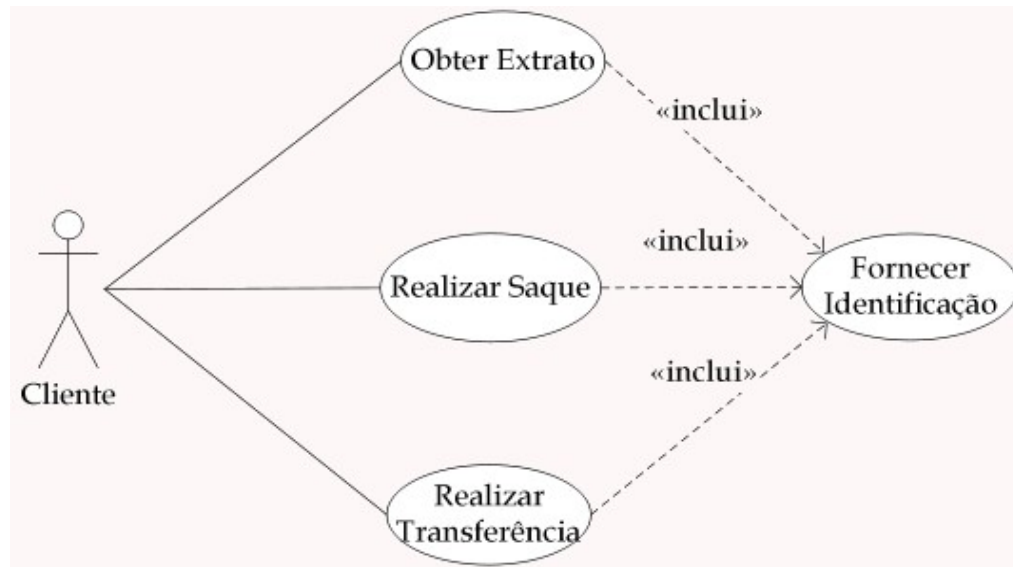


Diagrama de Caso de Uso (DCU)



Relacionamento de Inclusão

- ♦ Ocorre quando há uma parte de comportamento que é semelhante em mais de um caso de uso
- ♦ A seta (tracejada) de um relacionamento de inclusão recebe o estereótipo <<include>>
- ♦ **Só entre casos de uso**



Relacionamento de extensão

- ♦ Sejam A e B dois casos de uso.
 - Um relacionamento de extensão de A para B indica que um ou mais dos cenários de B podem incluir o comportamento especificado por A.
 - Neste caso, diz-se que B estende A.
 - O caso de uso A é chamado de estendido e o caso de uso B de extensor.
- ♦ Utilizado para modelar situações em que diferentes seqüências de interações podem ser inseridas em um caso de uso.
- ♦ Cada seqüência representa um comportamento **eventual**, que só ocorre sob certas condições ou cuja realização depende da escolha do ator.
- ♦ A existência do caso de uso estendido deve ser **independente** do caso de uso **extensor**

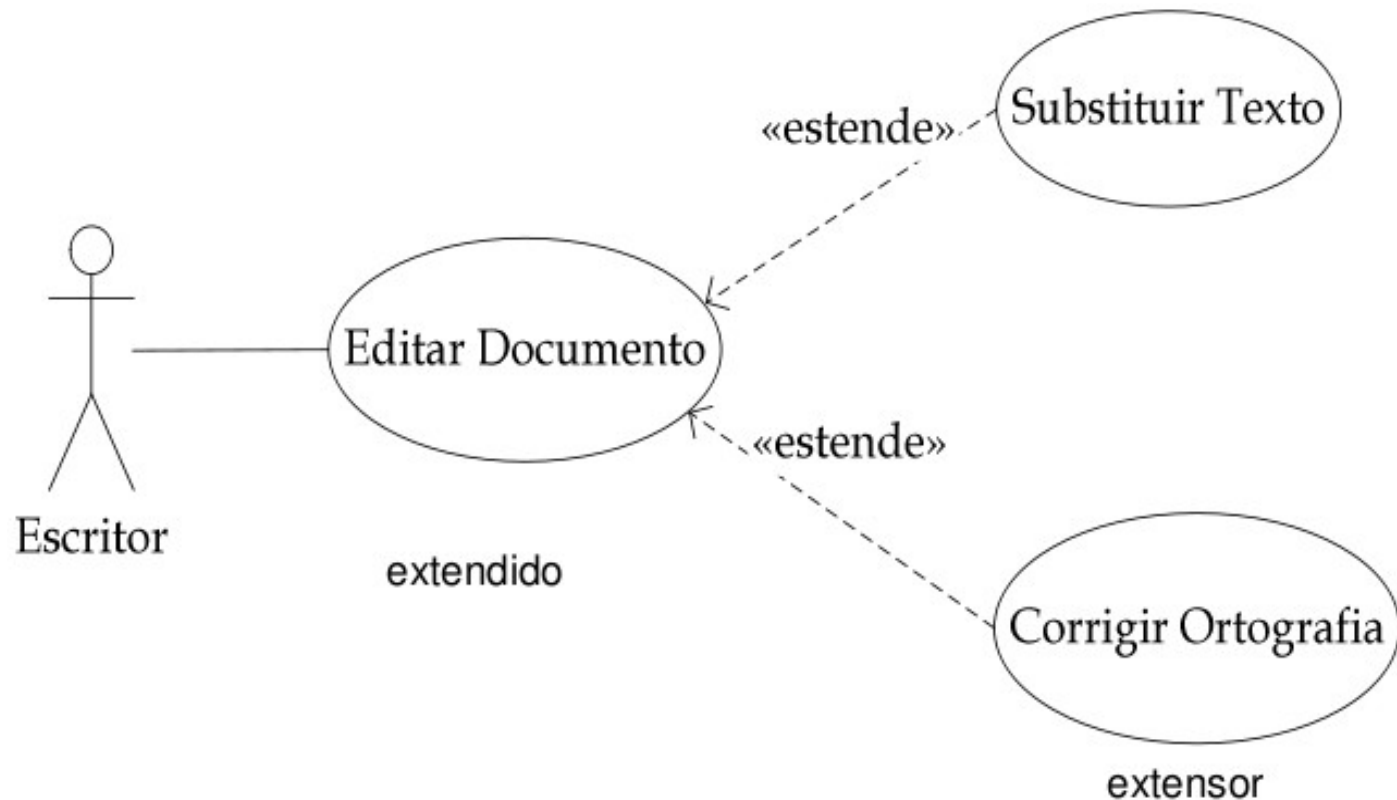
Relacionamento de extensão

- ◆ Exemplo: considere um processador de textos.
 - Caso de uso: **Editar Documento**.
 - No cenário típico deste caso de uso, o ator abre o documento, modifica-o, salva as modificações e fecha o documento.
 - Mas, em outro cenário, o ator pode desejar que o sistema faça uma **verificação ortográfica** no documento.
 - Em outro, o ator pode querer realizar a **substituição** de um fragmento de texto por outro.
 - Ambas extensões são **eventuais**
 - **Verificar Ortografia e Substituir Texto** estendem **Editar Documento**

Relacionamento de extensão

- ♦ Interações de **Substituir Texto**:
 - Em qualquer momento durante **Editar Documento**, o ator pode optar por substituir um fragmento de texto por outro.
 - O ator fornece o texto a ser substituído e o texto substituto.
 - O ator define os parâmetros de substituição (substituir somente palavras completas ou ocorrências dentro de palavras; substituir no documento todo ou somente na parte selecionada; ignorar ou considerar letras maiúsculas e minúsculas).
 - O sistema substitui todas as ocorrências encontradas no texto.
- ♦ A seta (tracejada) de um relacionamento de inclusão recebe o estereótipo <<**extends**>>.

Notação



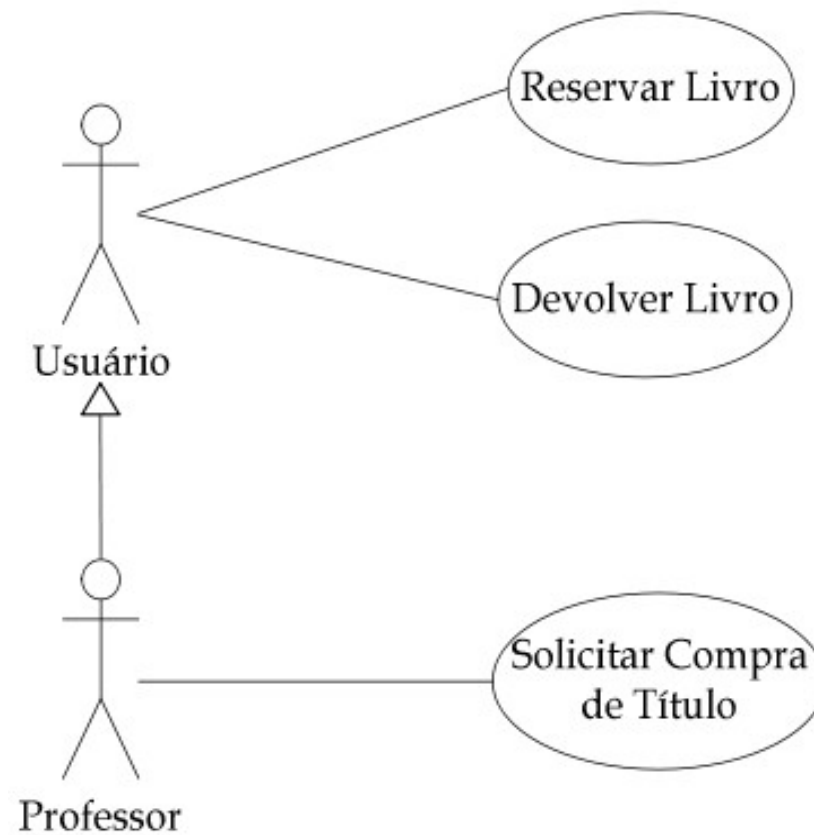
Relacionamento de Generalização

- ◆ Permite que um caso de uso (ou um ator) herde características de um caso de uso (ator) mais genérico.
- ◆ Pode existir entre **dois casos de uso** ou entre **dois atores**.
- ◆ O caso de uso (ator) herdeiro pode especializar o comportamento do caso de uso (ator) base.

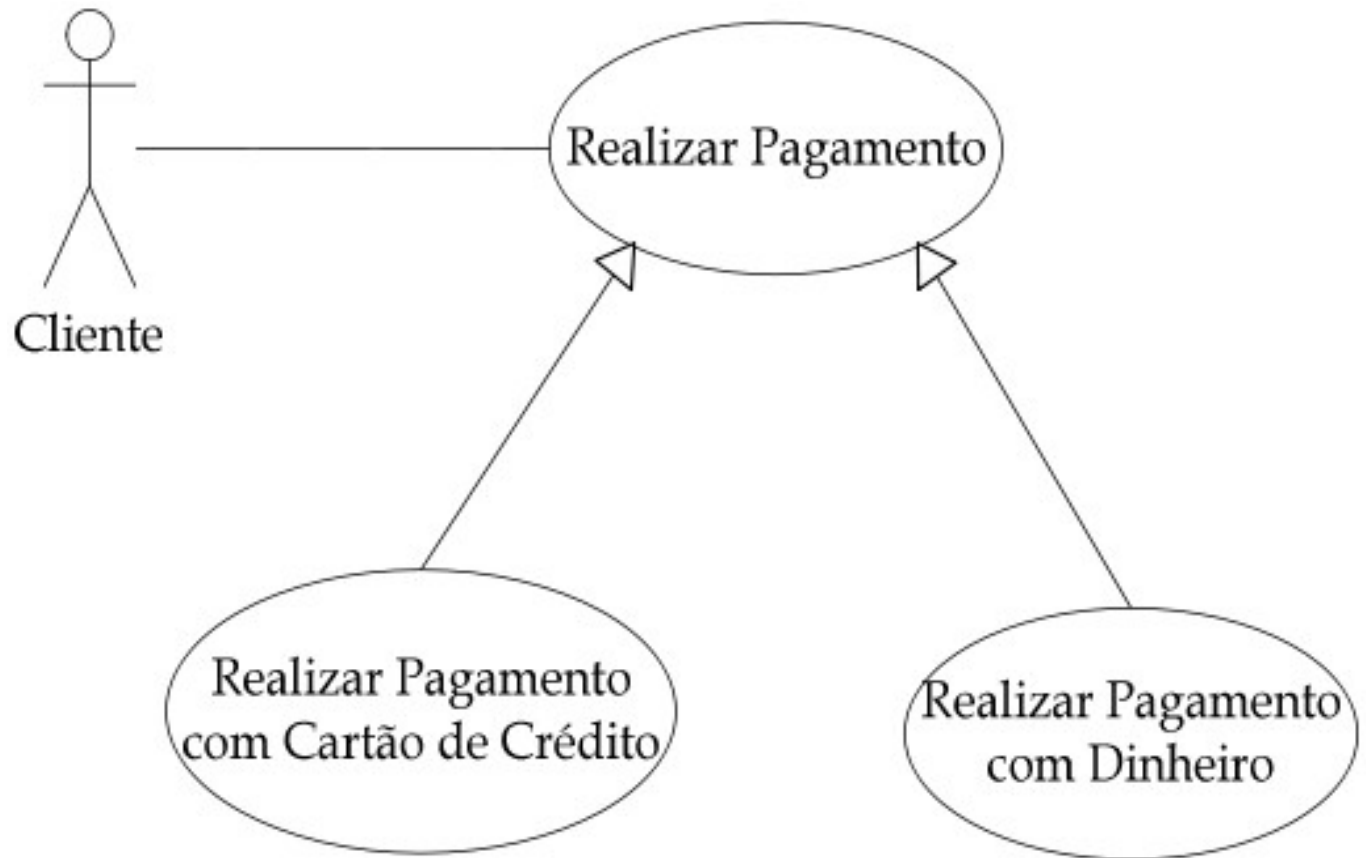
Relacionamento de Generalização

- ♦ Na **generalização entre casos de uso**, sejam A e B dois casos de uso.
 - Quando B herda de A, as seqüências de comportamento de A valem também para B.
 - Quando for necessário, B pode redefinir as seqüências de comportamento de A.
 - Além disso, B participa em qualquer relacionamento no qual A participa.
- ♦ Vantagem: comportamento do caso de uso original é reutilizado pelos casos de uso herdeiros.
 - Somente o comportamento que não faz sentido ou é diferente para um herdeiro precisa ser redefinido.

Notação



Notação





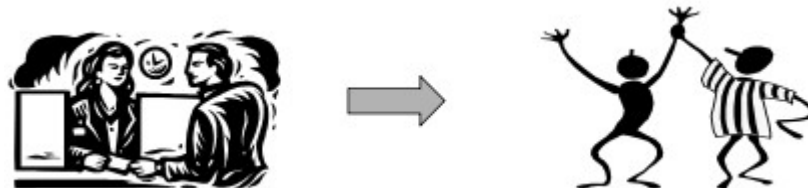
Identificação dos elementos do modelo de casos de uso

Identificação dos elementos do modelo

- ◆ Os atores e os casos de uso são identificados a partir do *levantamento de requisitos* do sistema
 - Analistas devem identificar as atividades dos processos de negócio relevantes ao sistema a ser construído
 - Identificar quais elementos que interagem nos processos
- ◆ **Quantos** casos de uso **são necessários** para descrever completamente um sistema?
 - Não há uma regra
 - Depende da complexidade do sistema.

Identificação dos atores

- ♦ O analista deve identificar:
 - as áreas da empresa que serão afetadas ou utilizarão o sistema
 - fontes de informações a serem processadas e os destinos das informações geradas pelo sistema.
 - Fontes e os destinos são **atores** em potencial



Identificação dos atores

- ◆ Perguntas úteis:
 - Que órgãos, empresas ou pessoas irão utilizar o sistema?
 - Que outros sistemas irão se comunicar com o sistema a ser construído?
 - Alguém deve ser informado de alguma ocorrência no sistema?
 - Quem está interessado em um certo requisito funcional do sistema?
- ◆ O desenvolvedor deve ainda continuar a pensar sobre atores quando passar para a identificação dos casos de uso.

Identificação de casos de uso

- ♦ A partir da lista (inicial) de atores, deve-se identificar os processos que eles iniciam ou dos quais eles participam (requisitos funcionais)
- ♦ Nessa identificação, pode-se distinguir entre dois tipos de casos de uso:
 - Primário: representa os objetivos dos atores.
 - Representam os processos que estão sendo automatizados.
 - Secundário: aquele que não traz benefício direto para os atores, mas que é necessário para que sistema funcione adequadamente.

Casos de uso primários

- ◆ Perguntas úteis:
 - Quais são as necessidades e objetivos de cada ator em relação ao sistema?
 - Que informações o sistema deve produzir?
 - O sistema deve realizar alguma ação que ocorre regularmente no tempo?
 - Para cada requisito funcional, existe um (ou mais) caso(s) de uso para atendê-lo?

Casos de uso secundários

- ◆ Estes se encaixam nas seguintes categorias:
 - Manutenção de cadastros.
 - Manutenção de usuários.
 - Manutenção de informações provenientes de outros sistemas.
- ◆ Importante: Um sistema de software não existe para cadastrar informações, nem tampouco para gerenciar os seus usuários.
 - O objetivo principal é produzir algo de valor para o ambiente no qual ele está implantado.
 - Começar a identificação dos casos de uso secundários é uma indicação de que o modelador está pensando em **como** o sistema dever ser construído.