



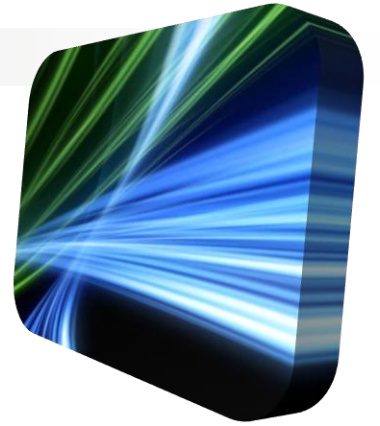
Engenharia de Software

Unified Modeling Language (UML)
Modelagem de Casos de Uso

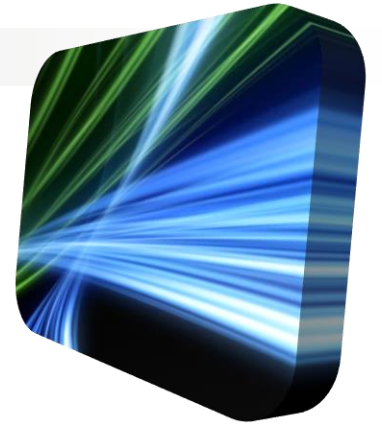
José Osvano da Silva, PMP

Sumário

- UML – Linguagem de Modelagem Unificada
- Modelo de Casos de Uso
- Casos de Uso
- Atores
- Diagrama de Casos de Uso
- Relacionamentos:
 - Include
 - Extend
 - Generalização



UML – Linguagem de Modelagem Unificada



- Principais autores do processo: Grady Booch, James Rumbaugh, Ivar Jacobson
- Chamados os 3 amigos
- Aproveitar o melhor das características das notações preexistentes
- *Notação da UML é uma união das diversas notações preexistentes com alguns elementos removidos e outros adicionados com o objetivo de torna-la mais expressiva.*

UML – Linguagem de Modelagem Unificada



- 1997 – A UML foi aprovada pela OMG (Object Management Group)
- A definição passa por constantes melhorias e conta com diversos colaboradores comerciais (Digital, HP, IBM, Oracle, Microsoft, Unysis, etc)
- 2003 – Foi lançada a UML 2.0
 - Especificação atual adotada pela OMG

UML – Linguagem de Modelagem Unificada



■ UML

- é uma linguagem visual para modelar sistemas Orientados a Objetos
- Define elementos gráficos que podem ser utilizados na modelagem de sistemas
- Através dos elementos definidos na linguagem podem-se construir diagramas para representar diferentes perspectivas de um sistema
- Cada elemento gráfico possui uma
 - Sintaxe: forma predeterminada de desenhar o elemento
 - Semântica: O que significa o elemento e com que objetivo deve ser usado
- A sintaxe e a semântica são extensíveis

UML – Linguagem de Modelagem Unificada



■ UML

- É independente de linguagens de programação e de processo de desenvolvimento
- Definição completa:
 - www.uml.org
 - Especificação de leitura complexa voltada a pesquisadores ou desenvolvedores de ferramentas de suporte

UML – Linguagem de Modelagem Unificada



■ Visões de um sistema

- Um sistema complexo pode ser examinado a partir de diversas perspectivas.
- Autores da UML definem 5 visões:
 - **Visão de Casos de uso:** Visão externa do sistema que define a interação entre o sistema e agentes externos.
 - **Visão de Projeto:** Características estruturais e comportamentais do sistema.
 - **Visão de Implementação:** gerenciamento de versões construídas pelo agrupamento de módulos e subsistemas.
 - **Visão de Implantação:** Distribuição física do sistema.
 - **Visão de Processo:** Características de concorrência, sincronização e desempenho do sistema.

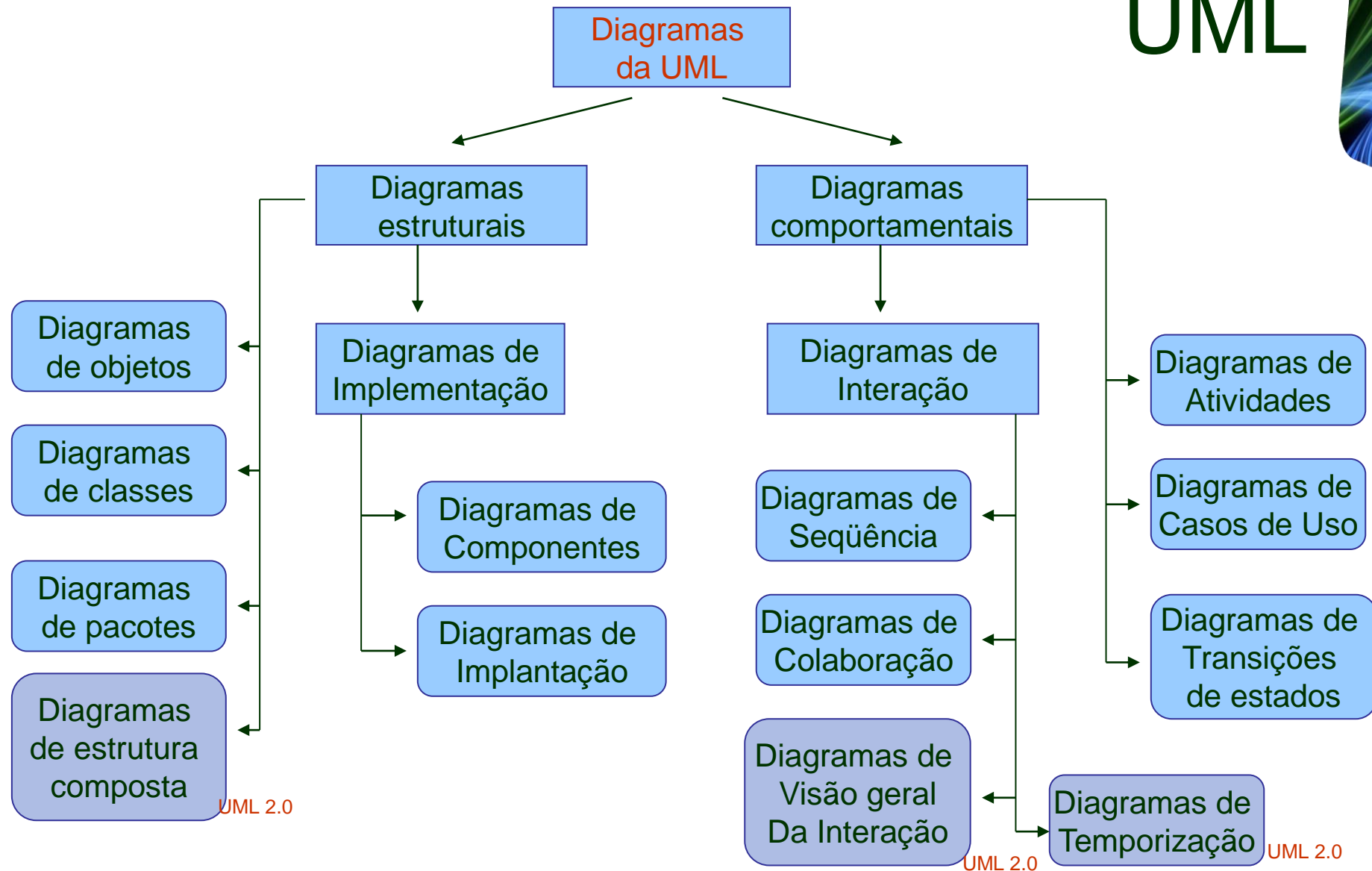
UML – Linguagem de Modelagem Unificada



■ Diagramas:

- Os documentos gerados em um processo de desenvolvimento são chamados de artefatos na UML
- Os artefatos compõe as visões do sistema
- A UML define 13 diagramas
- Esta quantidade de diagramas é justificada pela necessidade de analisar o sistema por meio de diferentes perspectivas
- Cada diagrama fornece uma perspectiva parcial do sistema.

UML



UML – Mecanismos gerais



■ Componentes da UML

- Blocos de construção básicos
- Regras que restringem como os blocos de construção podem ser associados
- Mecanismos de uso geral
 - Estereótipos, Notas explicativas, Etiquetas valoradas, Restrições, Pacotes, OCL

UML – Mecanismos gerais



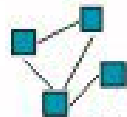
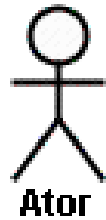
■ Estereótipos

- Estende o significado de determinado elemento em um diagrama
 - Existem estereótipos predefinidos
 - O usuário pode definir um estereótipo
- Um estereótipo deve ser documentado para evitar ambigüidades
- Estereótipos gráficos: Ícones gráficos
- Estereótipos textuais: Rótulo junto ao símbolo que representa.

UML – Mecanismos gerais



■ Estereótipos Gráficos



Process Model



Requirements

• Estereótipos Textuais

`<<document>>`
`<<interface>>`
`<<entity>>`

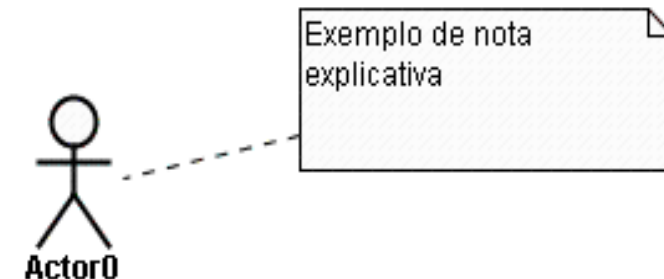
`<<satisfaz>>`
`<<realiza>>`

UML – Mecanismos gerais



■ Notas explicativas

- Comenta ou esclarece alguma parte do diagrama
 - Textuais
 - Linguagem de restrição de objetos (OCL)
- Não modificam nem estendem o significado do elemento
- Não deve ser usado em excesso

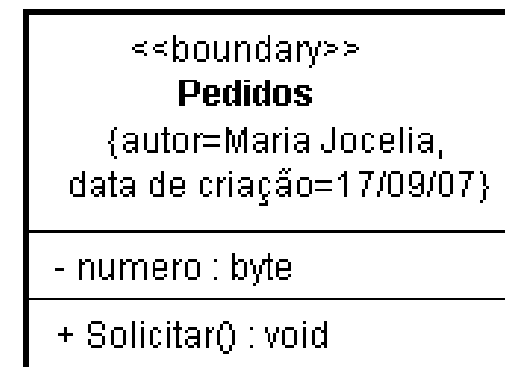


UML – Mecanismos gerais



■ Etiquetas valoradas (tagged value)

- Os elementos da UML tem 3 propriedades predefinidas: nome, lista de atributos e lista de operações
- Etiquetas valoradas são usadas para definição de outras propriedades além das 3 predefinidas
- Na UML 2.0 somente pode-se usar uma etiqueta valorada como um atributo usado sobre um estereótipo
- Notação
 - {tag=valor}

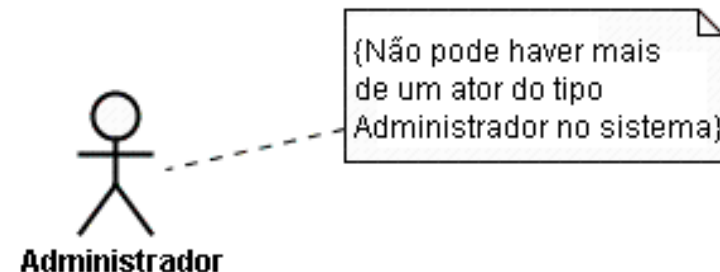


UML – Mecanismos gerais



■ Restrições

- Podem estender ou alterar a semântica natural de um elemento gráfico
- Podem ser especificadas formalmente (OCL) ou informalmente (texto livre)
- Restrições devem aparecer dentro de notas explicativas



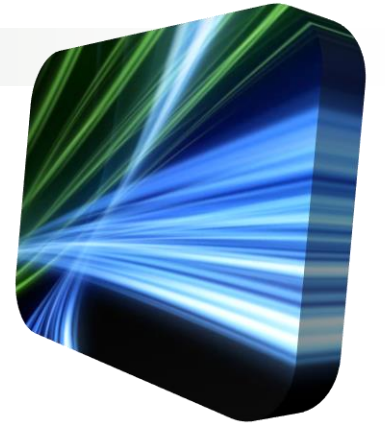
UML – Mecanismos gerais



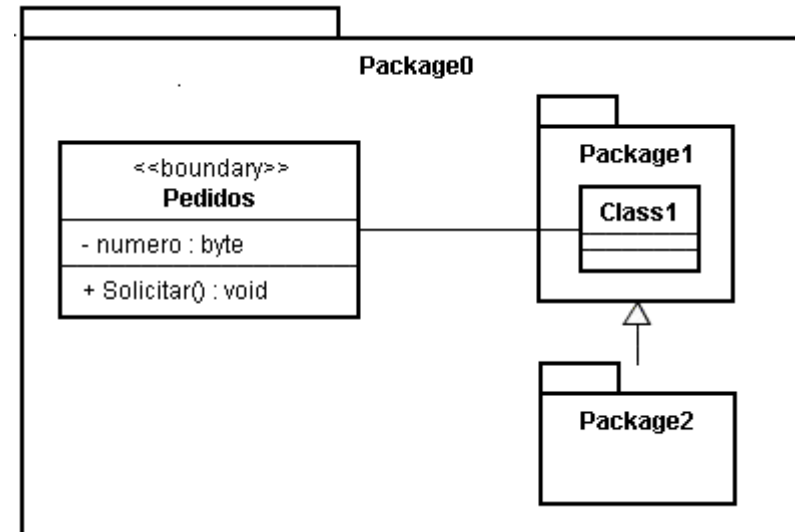
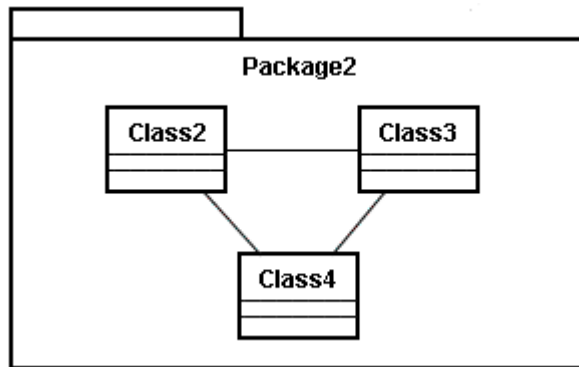
■ Pacotes

- Agrupa elementos semanticamente relacionados
- Um pacote se liga a outro através de um relacionamento de dependência
- A dependência pode ser especificada através de um estereótipo
- Pode agrupar outros pacotes

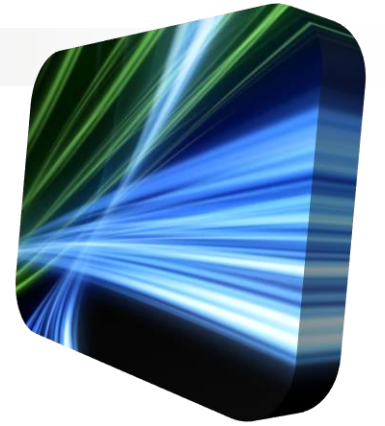
UML – Mecanismos gerais



■ Pacotes



UML – Mecanismos gerais



■ OCL (Linguagem de restrição de objetos)

- Linguagem formal para especificar restrições sobre diversos elementos em um modelo
- Consiste de:
 - Contexto: Domínio no qual a declaração em OCL se aplica
 - Propriedade: um componente do contexto
 - Operação: O que deve ser aplicado sobre a propriedade
- Exemplo:

Veículo
- proprietario : Pessoa
- cor : String
- marca : String

Context Veículo

inv: `self.proprietário.idade >= 18`

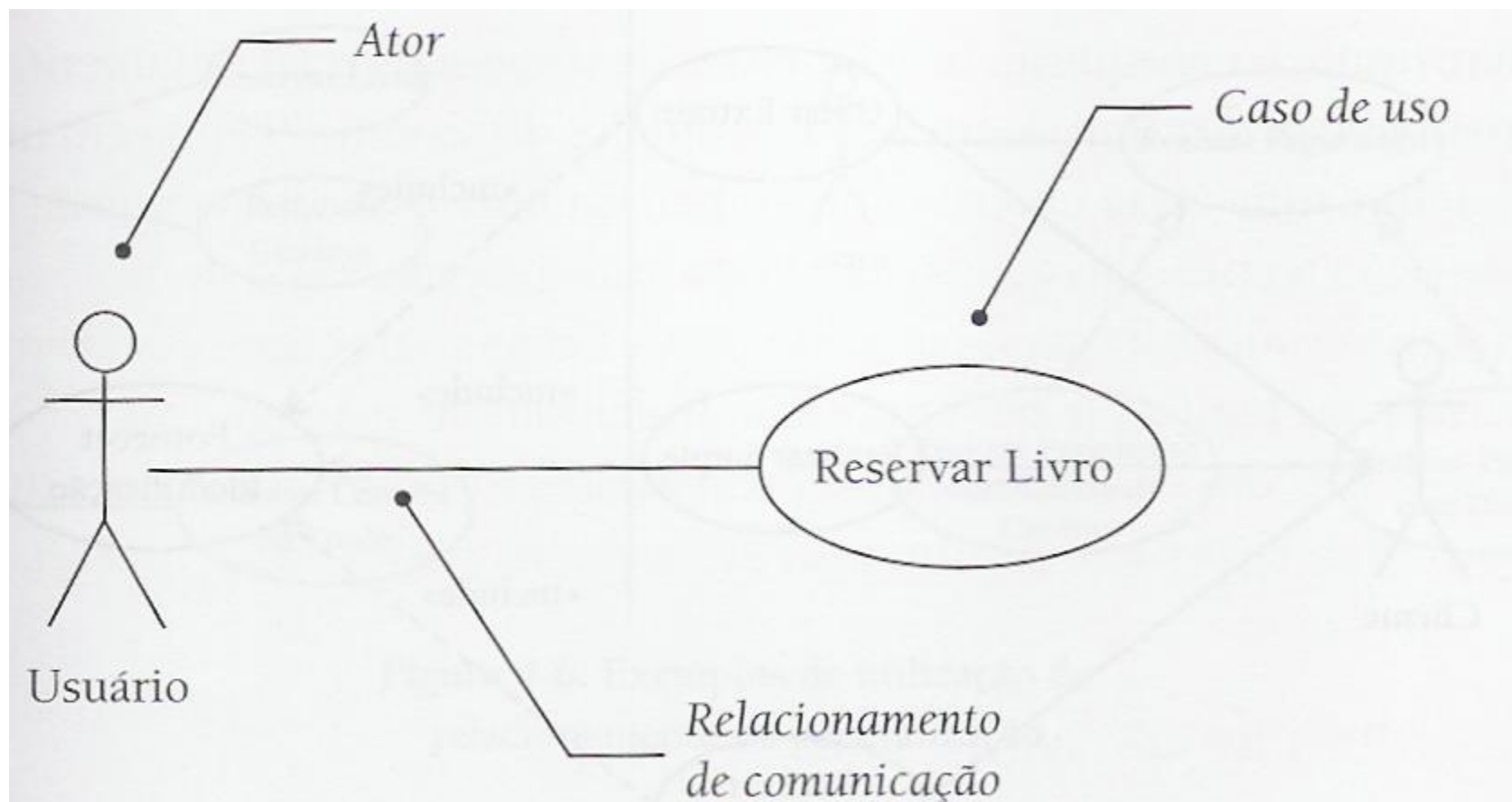
Modelos de Casos de Uso



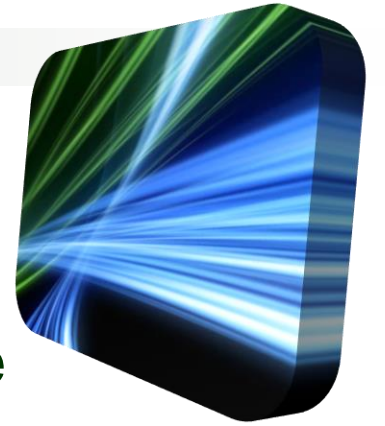
- ♦ Introdução

- ♦ É uma representação das funcionalidades eternamente observáveis do sistema e dos elementos externos ao sistema que interagem com eles
- ♦ É um modelo de análise que representa um refinamento dos requisitos funcionais.
- ♦ Idealizado por Ivar Jacobson em 1970 e inserida na UML na década de 90.
- ♦ É o modelo mais popular para a documentação de requisitos funcionais
- ♦ O MCU representa os possíveis usos de um sistema.
- ♦ Componentes: *Casos de Usos, Atores e Relacionamentos*

Notação da UML



Casos de Uso

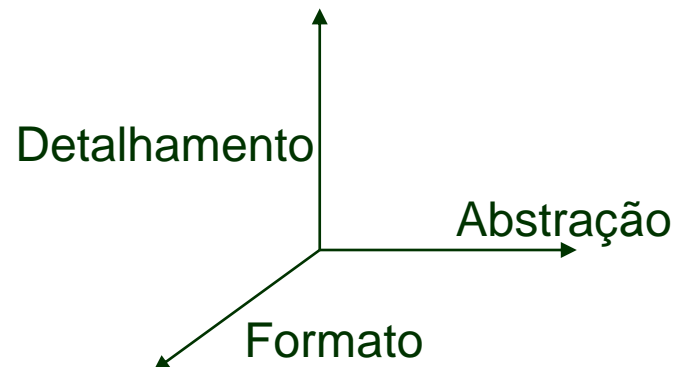


- ♦ É a especificação de uma **sequência completa** de interações entre um **sistema** e um ou mais **agentes externos** a este sistema.
 - ♦ Representa uma determinada funcionalidade de um sistema conforme percebida externamente.
 - ♦ Representa também os agentes externos que interagem com o sistema
 - ♦ Não revela a estrutura e o comportamento interno do sistema.
- ♦ Completa representa um **relato fim a fim** de um dos usos do sistema para alcançar um objetivo útil.
 - ♦ *Ex: Entrar no sistema **não** é um caso de uso*

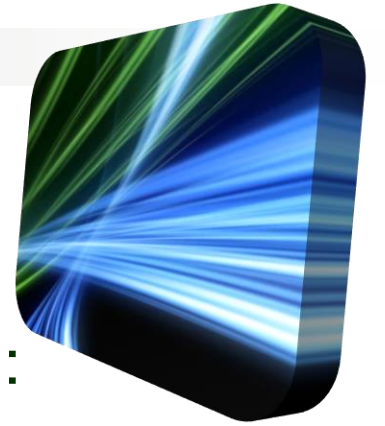
Casos de Uso



- ♦ Um MCU pode conter vários casos de uso
- ♦ Cada caso de uso se define pela descrição narrativa das interações entre o agente externo e o sistema.
- ♦ Há 3 dimensões para variações das descrições dos casos de uso



Casos de Uso



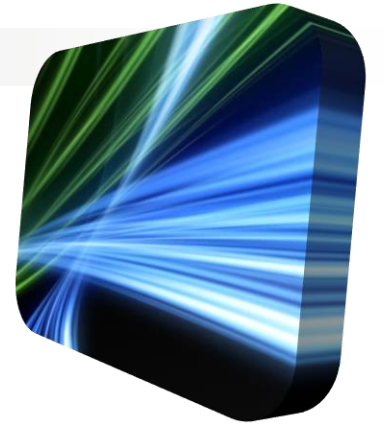
- ♦ Formato: estrutura utilizada para organizar a sua narrativa textual:
 - ♦ Contínuo, numerado, tabular
- ♦ Formato contínuo

Este caso de uso inicia quando 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 Cliente fornece o valor da quantidade que deseja sacar. O sistema fornece a quantia desejada e imprime o recibo para o Cliente. O Cliente retira a quantia e o recibo, e o caso de uso termina.

Casos de Uso

- ◆ Formato numerado

- 1) Cliente insere seu cartão no caixa eletrônico.
- 2) O sistema requisita a senha do Cliente.
- 3) Cliente fornecer sua senha
- 4) Sistema valida a senha e exibe as opções de operações possíveis.
- 5) O Cliente opta por realizar um saque.
- 6) Sistema requisita o total a ser sacado.
- 7) O Cliente fornece o valor da quantidade que deseja sacar.
- 8) O sistema fornece a quantia desejada e imprime o recibo para o Cliente.
- 9) O Cliente retira a quantia e o recibo , e o caso de usa termina.



Casos de Uso

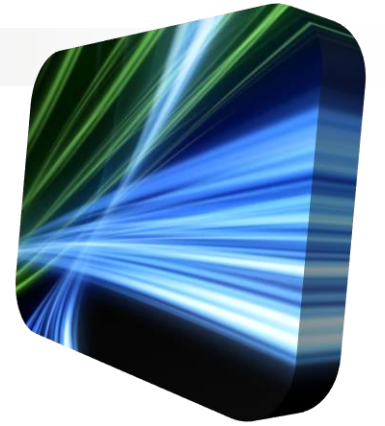


- ◆ Formato Tabular

Cliente	Sistema
Insere seu cartão no caixa eletrônico.	Requisita a senha do Cliente.
Fornecer sua senha	Valida a senha e exibe as opções de operações possíveis.
Solicita realização de um saque.	Requisita o total a ser sacado.
Fornece o valor da quantidade que deseja sacar.	Fornece a quantia desejada e imprime o recibo para o Cliente.
Retira a quantia e o recibo	

Tenta prover alguma estrutura à descrição

Casos de Uso



- ♦ Grau de detalhamento
 - ♦ Sucinto: Não detalha as interações
 - ♦ Expandido: Descreve as interações em detalhes
- ♦ Grau de abstração
 - ♦ Existência ou não de menção a aspectos relativos a tecnologia durante a descrição de um caso de uso
 - ♦ Real: Se compromete com a solução do projeto
 - ♦ Ex: O usuário insere o seu cartão magnético
 - ♦ Essencial: É abstrato no sentido de não mencionar aspectos relativo ao uso de tecnologias
 - ♦ Ex: O usuário fornece sua identificação

Casos de Uso



◆ Cenários

- ◆ É a descrição de uma das maneiras pelas quais o caso de uso pode ser utilizada.
- ◆ É um episódio de utilização de uma funcionalidade.
- ◆ Pode ser utilizada posteriormente na fase de testes
- ◆ Pode ser vista como uma instância de um caso de uso.

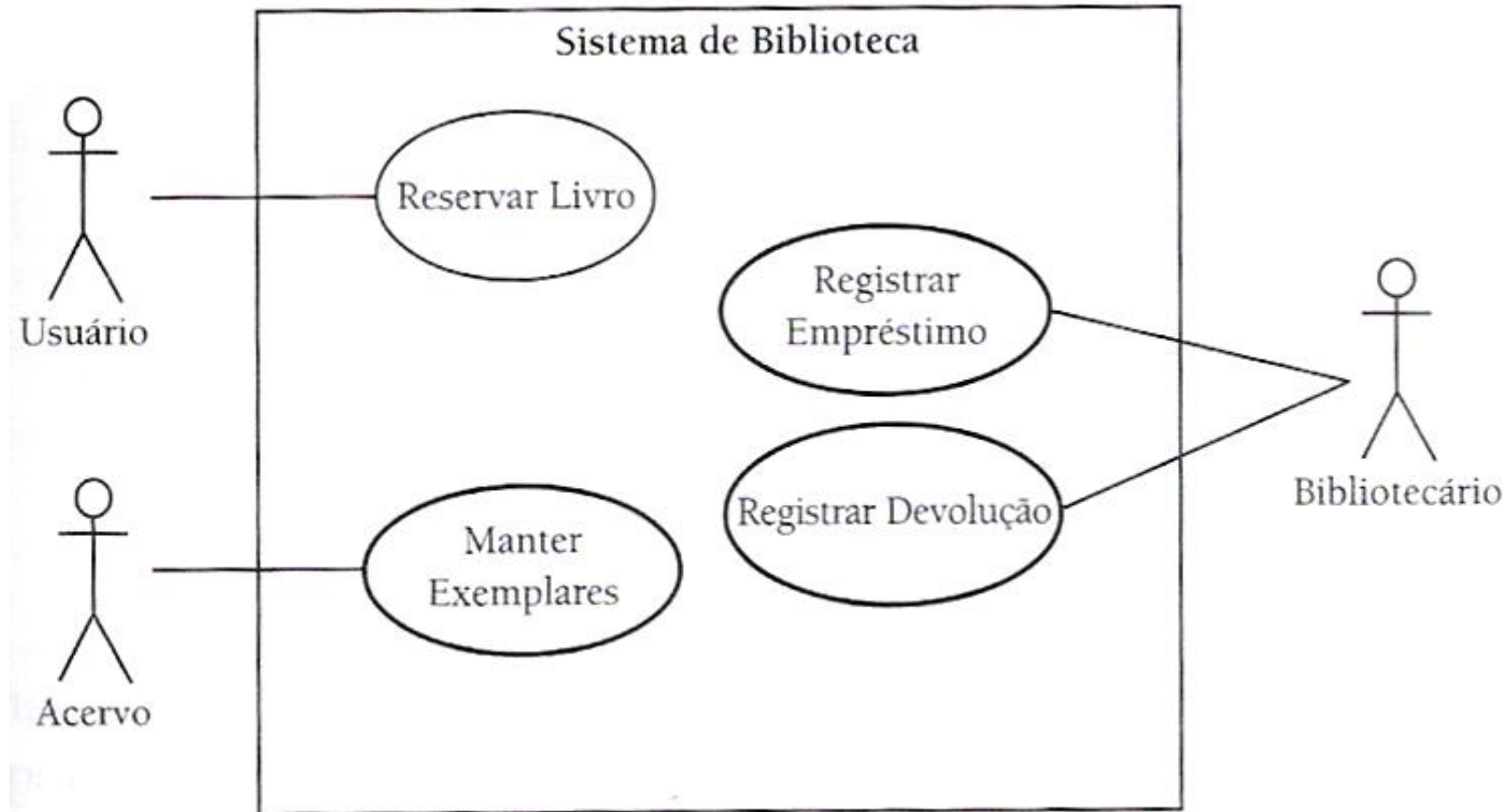
- Cliente insere seu cartão no caixa eletrônico.
- O sistema apresenta a tela de requisição de senha do Cliente.
- Cliente digita sua senha
- Sistema valida a senha e exibe o menu com as opções de saque, pagamento ou transferência.
- O Cliente seleciona a opção saque.
- Sistema apresenta tela com a requisição do valor a ser sacado.
- O Cliente digita o valor da quantidade que deseja sacar.
- ...

Atores

- ♦ É qualquer elemento externo ao sistema que interage com o mesmo
 - ♦ Atores não fazem parte do sistema
 - ♦ Atores trocam informações com o sistema
- ♦ Um ator representa um papel representado em relação ao sistema
- ♦ Categorias
 - ♦ Cargos
 - ♦ Organizações ou divisões de uma organização
 - ♦ Outros sistemas de software
 - ♦ Equipamentos que o sistema se comunica
- ♦ Atores podem ser Primários ou Secundários



Diagrama de Casos de Uso



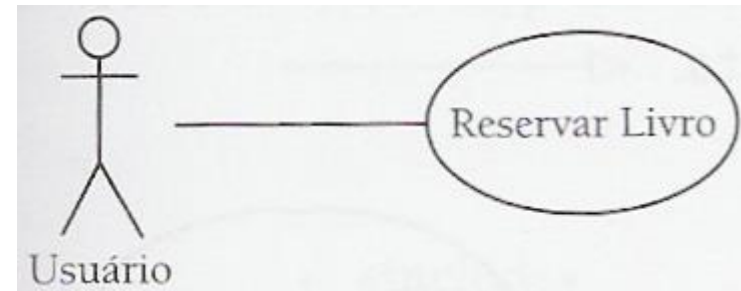
Relacionamentos



- ♦ Componente responsável por representar a interação entre os atores e casos de usos. (**Ator <--> Caso de Uso**)
- ♦ Também representa ligações entre casos de uso ou entre atores. (**Ator <--> Ator; Caso de Uso <--> Caso de Uso**)
- ♦ Tipos de Relacionamentos no MCU:
 - ♦ Comunicação
 - ♦ Inclusão
 - ♦ Extensão
 - ♦ Generalização

Relacionamentos

- ♦ Comunicação:
 - ♦ Informa a que caso de uso o ator está associado
 - ♦ Representa as trocas de informação entre os atores e casos de uso
 - ♦ É o mais utilizado nos MCU
 - ♦ Um ator pode estar associado a vários casos de uso
 - ♦ Um caso de uso pode estar associado a vários atores

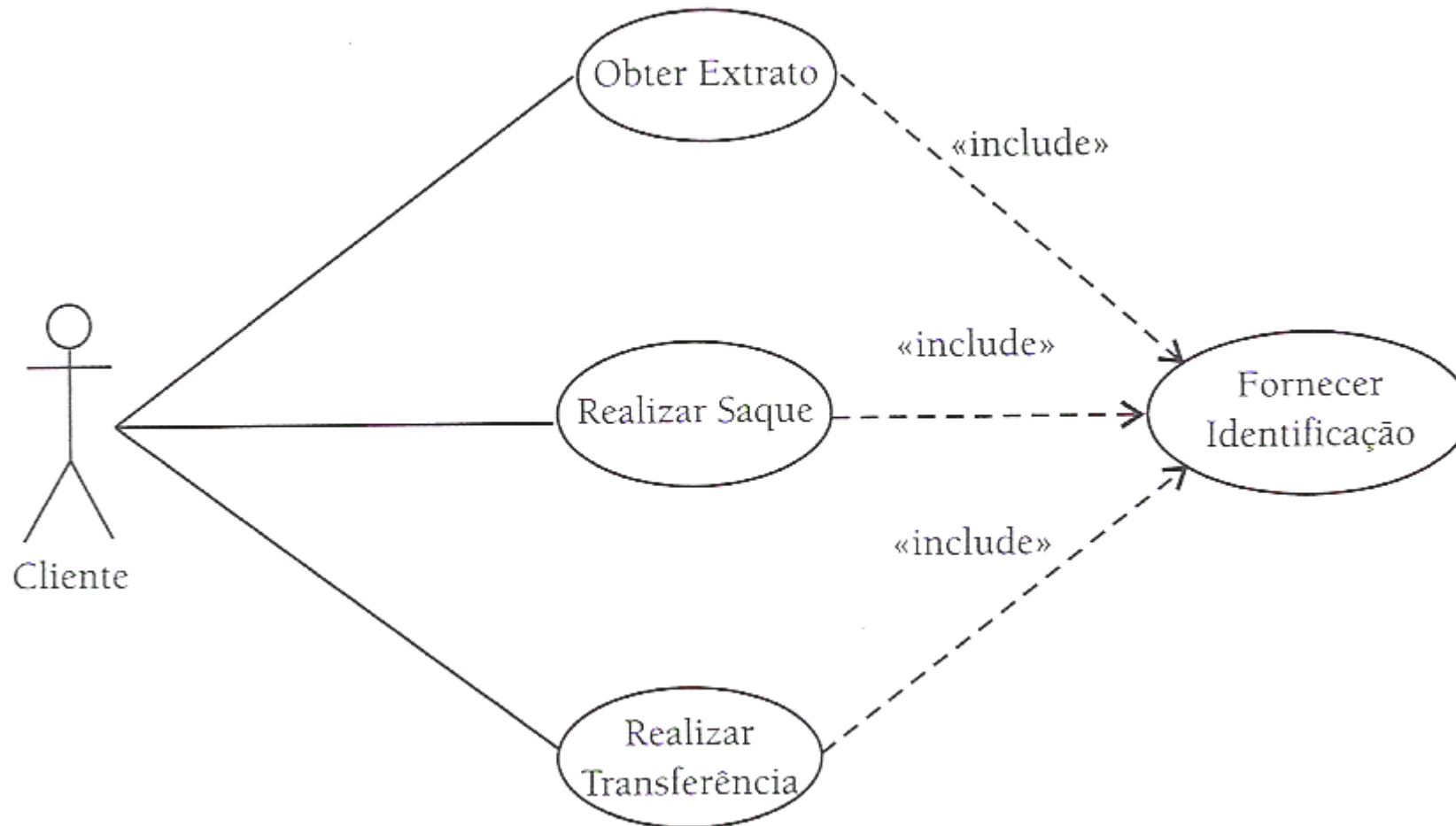


Relacionamentos

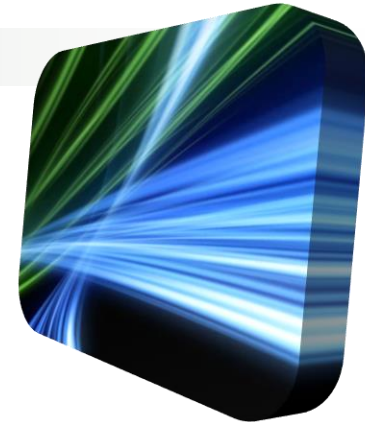


- ◆ Inclusão:
 - ◆ Somente entre Casos de Usos
 - ◆ Quando dois ou mais casos de usos incluem uma sequencia comum de interações, esta sequencia pode ser descrita em outro caso de uso
 - ◆ Vários caos de uso podem incluir o comportamento deste caso de uso comum.
 - ◆ Ex: *Obter Extrato, Realizar Saque e Transferência* incluem *Validar Senha*

Diagrama de caso de Uso - Inclusão

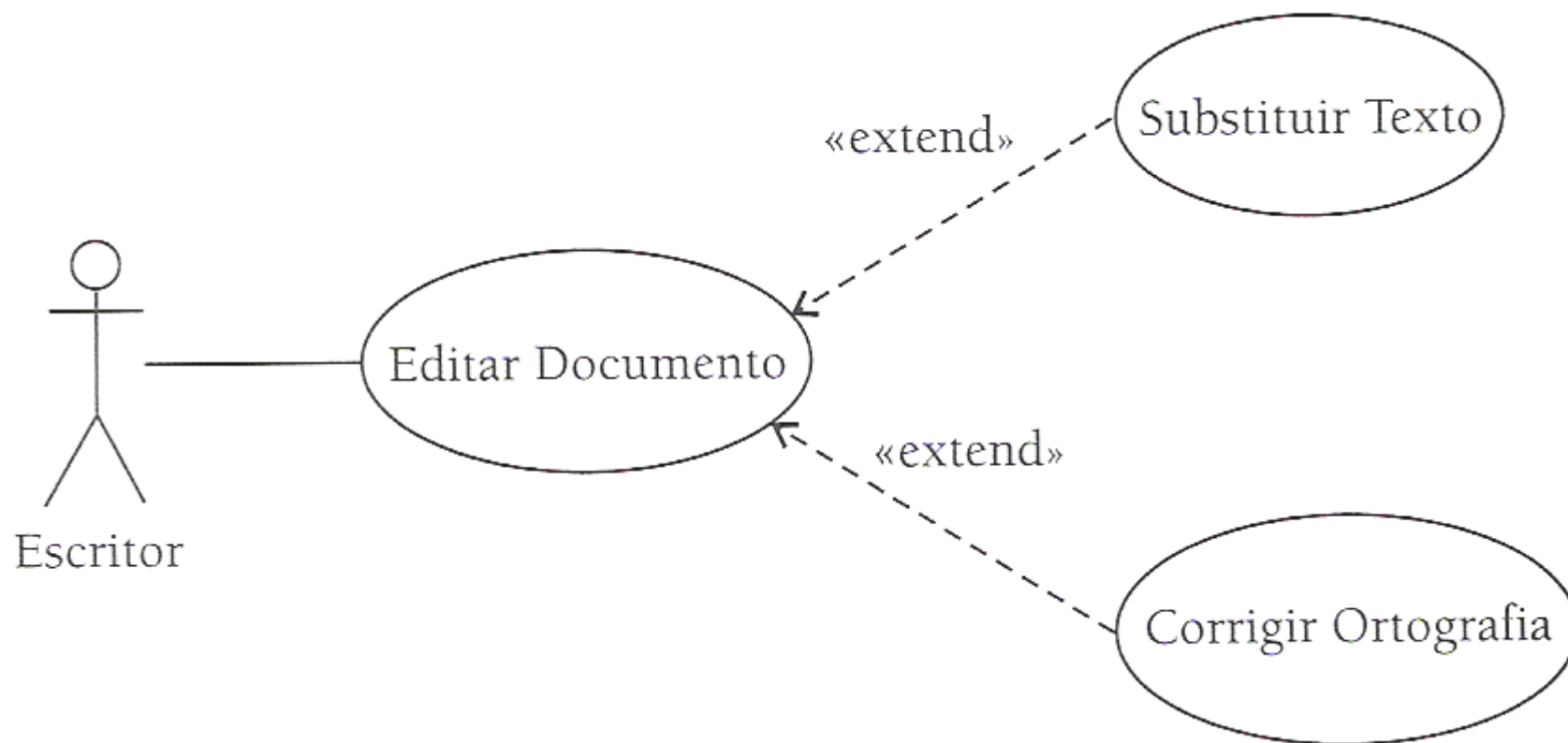
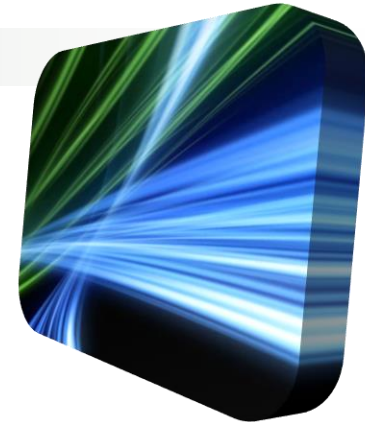


Relacionamentos

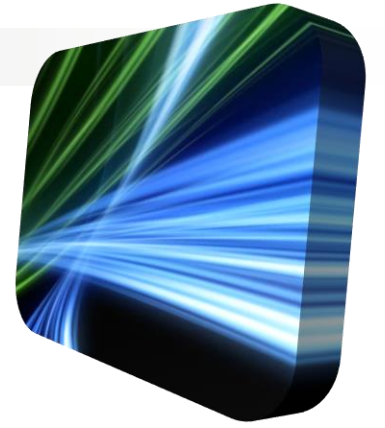


- ◆ Extensão:
 - ◆ Somente entre Casos de Usos
 - ◆ Modelar situações em que diferentes seqüências de interações podem ser inseridas em um mesmo caso de uso. Estas seqüências representam um **comportamento eventual**.
 - ◆ A existência de um caso de uso estendido deve ser **independente** da existência de casos de uso que estendam o primeiro
 - ◆ Exemplo: *Realizar Saque e Transferência* podem ser **estendidos** por *Consultar Extrato*

Diagramas de Caso de Uso - Extensão



Relacionamentos



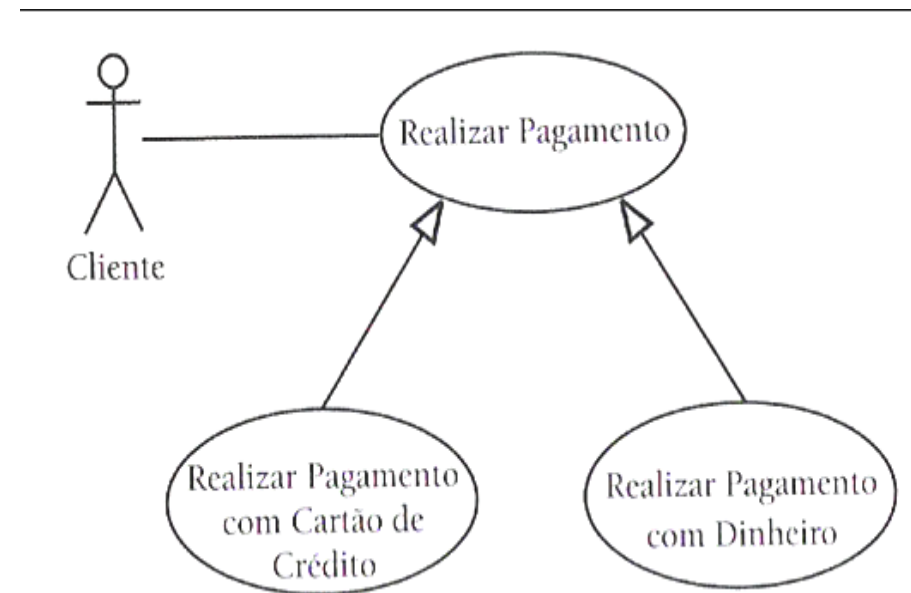
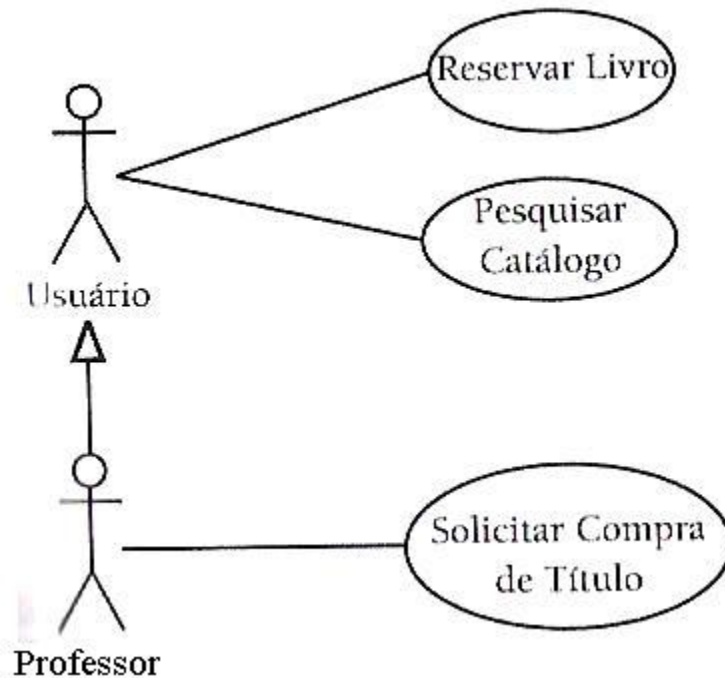
- ♦ Generalização:
 - ♦ Pode existir entre dois casos de Uso ou entre dois atores
 - ♦ Permite que um caso de uso (ou um ator) herde comportamento de outro caso de uso (ou ator)
 - ♦ O caso de uso mais genérico pode ser Abstrato ou concreto.
 - ♦ É recomendado que o caso de uso pai sempre seja abstrato para evitar problemas na especificação
 - ♦ O caso de uso pai é utilizado apenas para representar a natureza dos casos de uso filho.
 - ♦ Não há especificação de comportamento para o caso de uso abstrato.

Relacionamentos

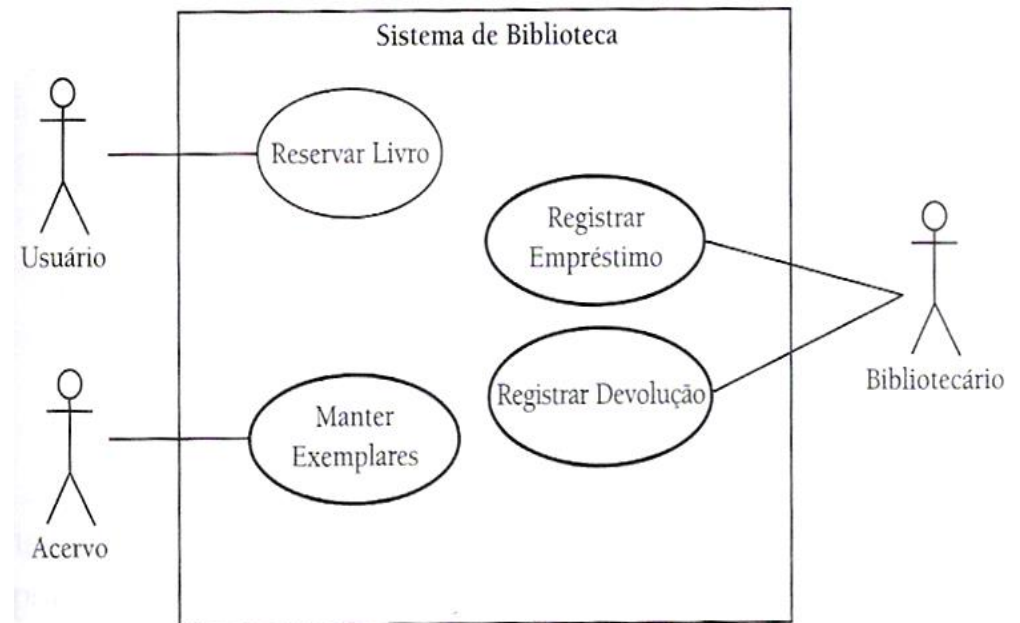
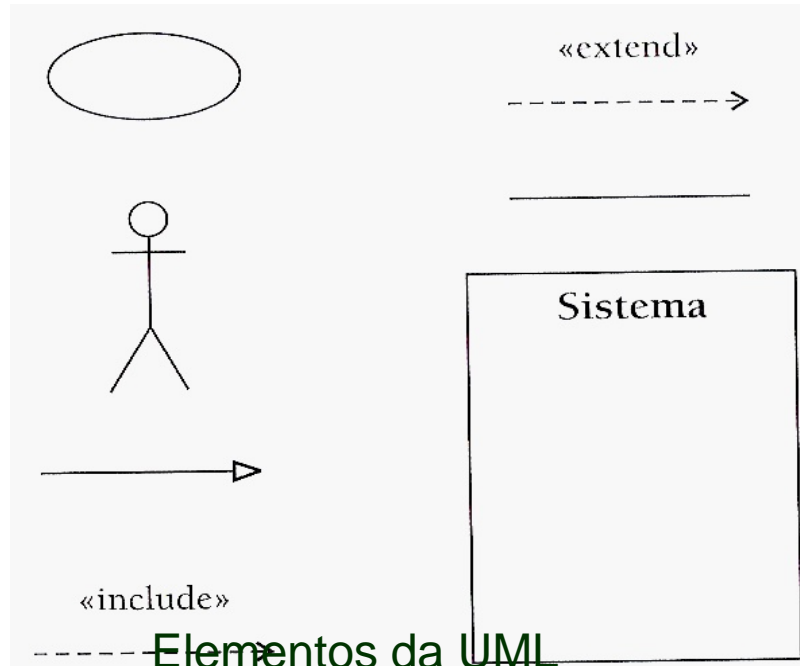


- ♦ Generalização:
 - ♦ Exemplos:
 - ♦ *Requisitar Produtos é generalização de Requisitar produtos na loja, Requisitar produtos pela internet*
 - ♦ *Usuário é generalização de Professor e Aluno*

Diagramas de Caso de Uso - Generalização



Diagramas de Casos de Uso



Quando usar relacionamentos



- ♦ Inclusão
 - ♦ Quando o mesmo comportamento se repete em mais de um caso de uso
 - ♦ O Comportamento comum necessariamente contido em todos os cenários do caso de uso *inclusor*
 - ♦ O caso de uso *inclusor* não está completo sem o caso de uso *incluso*
- ♦ Extensão
 - ♦ Quando um comportamento eventual do caso de uso tiver que ser descrito
 - ♦ Para estender o comportamento do caso de uso sem modificar o original

Quando usar relacionamentos



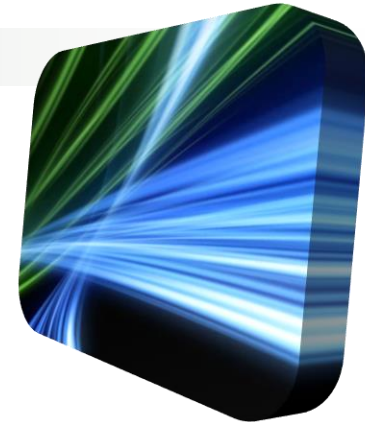
- ◆ Generalização de caso de uso
 - ◆ Identifica-se vários casos de uso com o mesmo comportamento
 - ◆ Se o comportamento do pai difere em alguma coisa do do filho, não use generalização, mas extensão.
- ◆ Generalização de Ator
 - ◆ Precisa definir um ator que desempenha papel já desempenhado por outro ator em relação ao sistema, mas que também possui comportamento particular
- ◆ A legibilidade tem preferência sobre a formalização
 - ◆ Nunca use muitos relacionamentos de extensão, inclusão e generalização

Referências



- Craig Larman, 2007, “**Utilizando UML e Padrões**”, 3ª ed.
- SOMMERVILLE, Ian, **Engenharia de Software**, 8ª Edição, São Paulo, Editora Pearson Prentice Hall, 2007.
- PRESSMAN, R. S. **Engenharia de software**: uma abordagem profissional. 7ª Edição. Porto Alegre: AMGH, 2011. 780 p.

Dúvidas



José Osvano da Silva
 joseosvano@unipac.br