



UNIVERSIDADE FEDERAL DO PIAUÍ – UFPI
CAMPUS SENADOR HELVÍDIO NUNES DE BARROS - CSHNB
CURSO DE BACHARELADO EM SISTEMAS DE INFORMAÇÃO



Engenharia de Software I

Introdução a UML

Professora Pâmela Carvalho

07/02/23

Introdução a UML

O que é um modelo?



O que é um modelo?

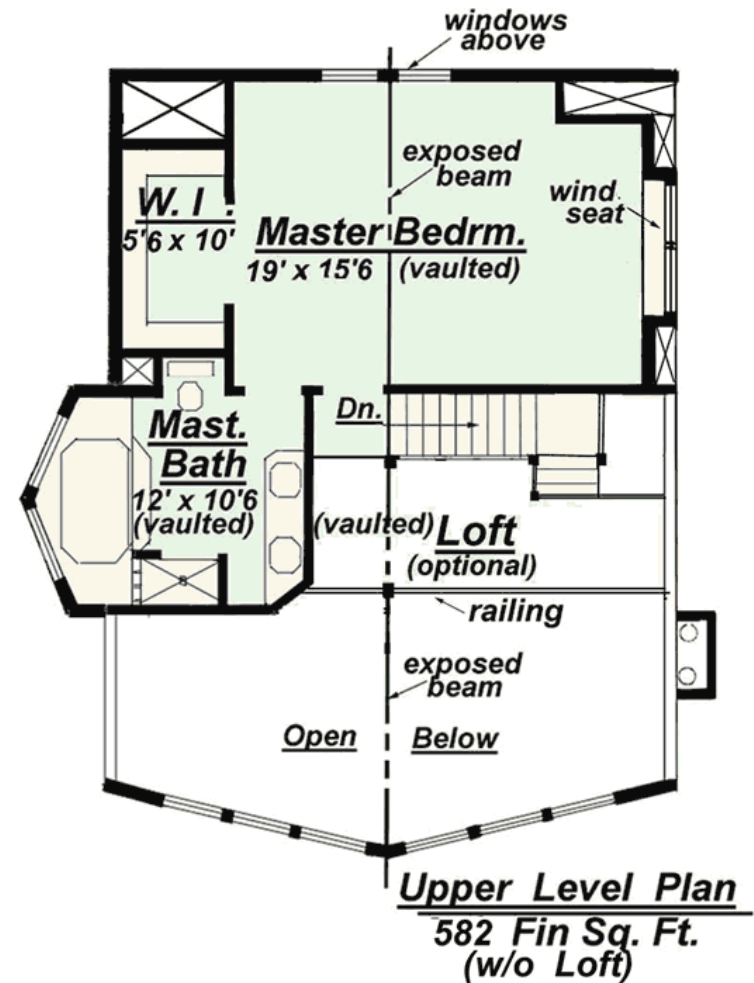
- Um modelo é uma **simplificação** da realidade;
- Nesse contexto, construímos modelos para **compreender** melhor o sistema que estamos desenvolvendo;



O que é um modelo?



O que é um modelo?



Porque modelar um software?



Muitos profissionais afirmam que conseguem determinar as necessidades do sistema “de cabeça”.

ALGO ERRADO! Ex: Um engenheiro/pedreiro dificilmente conseguirá construir uma casa (com qualidade) sem um projeto.

Essa modelagem...

- Ajuda a ter uma **visão geral do sistema**;
- Permite **especificar a estrutura e o comportamento** do sistema;
- Proporciona um **guia para a construção** do sistema;
- **Documenta as decisões** tomadas.



O que é UML?



- A UML é uma **linguagem visual** utilizada para modelar sistemas computacionais por meio do **Paradigma de Orientação a Objetos**.
- Tem como **objetivo auxiliar na definição das características** do software ou dos elementos inerente ao desenvolvimento do mesmo.
- UML sozinha não resolve...
 - Ela deve ser usada dentro de um processo de desenvolvimento.

O que é UML?



- ...uma **linguagem gráfica** para visualizar, especificar, construir e documentar os **artefatos de um sistema** de software.
- ...adotada por grande parte da indústria de software e por fornecedores de ferramentas CASE como **linguagem padrão de modelagem**.
- ...utilizada com **qualquer** processo de desenvolvimento

O que é UML?



- 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

– É independente de linguagens de programação e de processo de desenvolvimento.

– Definição completa:

- www.uml.org

Criadores da UML

- ❖ James Rumbaugh - Object Modeling Technique (OMT)
- ❖ Grady Booch - Booch Method
- ❖ Ivar Jacobson - Objectory (OOSE) Process



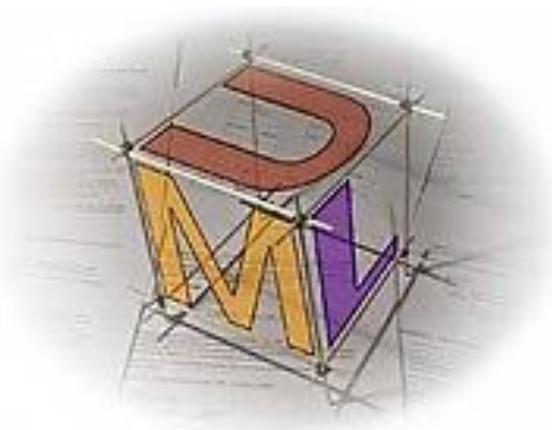
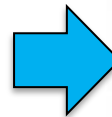
G. Booch



I. Jacobson



J. Rumbaugh



- Cada autor adotava ideias dos métodos dos outros, então, evoluindo juntos produziram melhorias.
- A unificação dos 3 métodos traria estabilidade para o mercado.

Histórico da UML

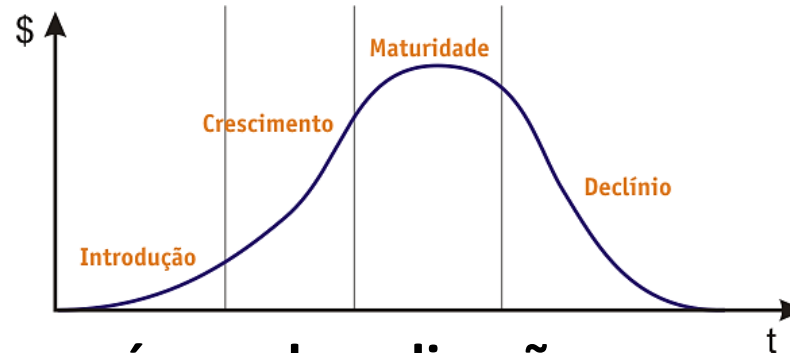
- **Outubro/1994: Booch e Jim Rumbaugh** começaram um esforço para unificar o método de **Booch** e **OMT** (*Object Modeling Language*).
- **Primeira versão, chamada *Unified Method***, foi divulgada em outubro/1995.
- **Jacobson** juntou-se ao grupo, agregando o método **OOSE** (*Object-Oriented Software Engineering*).
- **Esforço dos três resultou na liberação da UML v. 0.9 e 0.91 em junho e outubro/1996. Em janeiro/1997, foi liberada a versão 1.0 da UML.**
- **Adotada como padrão segundo a OMG (*Object Management Group*, <http://www.omg.org/>) em Novembro/1997**
- **UML 2.0 em 2004 e UML 2.5 (versão atual) em 2015**

Informações extras sobre UML



Informações extras sobre UML

- Suporta todo o **ciclo de vida** do software;



- Suporta **diversas áreas de aplicação**;
- É **baseado na experiência e necessidades** da comunidade de utilizadores;
- É suportado por muitas **ferramentas**.

UML é uma **linguagem de modelagem** para:

- Visualização
- Especificação
- Construção
- Documentação

Informações extras sobre UML

○ Visualização:

- A existência de um **modelo visual facilita a comunicação** e faz com que os membros de um grupo tenham a mesma ideia do sistema.
- Cada símbolo gráfico tem uma semântica bem definida.

○ Especificação:

- Construir **modelos precisos, sem ambiguidades e completos**.
- A UML atende todos os requisitos de especificação dentro de um processo, desde a fase de análise até a fase de testes e implementação do sistema concluído.

○ Construção:

- Na UML é possível realizar um **mapeamento dos modelos gerados**, para as linguagens de programação e até mesmo para banco de dados relacionais ou orientados a objetos.

○ Documentação:

- Artefatos como requisições de negócios, modelo de arquitetura, código fonte, modelo de análise, protótipo e outros documentos, pode ser **documentados com a UML**.

Informações extras sobre UML

- **Visões de um sistema**

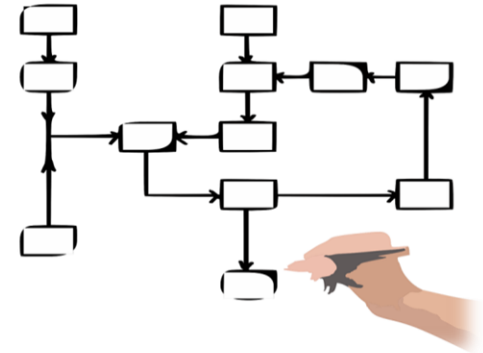
- Um sistema complexo pode ser examinado a partir de diversas perspectivas.
- Autores da UML definem 5 visões:
 - **Visão de Casos de uso**: Visão externa do sistema que define a interação entre o sistema e agentes externos.
 - **Visão de Projeto**: Características estruturais e comportamentais do sistema.
 - **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.

Porque usar UML?



- É padronizado (**garante organização**).
- Para **comunicar a estrutura e o comportamento desejado** de um sistema.
- Para **visualizar e controlar a arquitetura** de um sistema.
- Para **melhorar o nosso entendimento de um sistema** e, assim, expor oportunidades para melhorias e reutilização.
- Para utilização de uma **notação padronizada** que abrange qualquer tipo de sistema.
- Pela facilidade **no entendimento da orientação a objetos**.
- **Conceito em realidade**.

Porque usar diagramas UML?



○ Os diagramas UML são usados para:

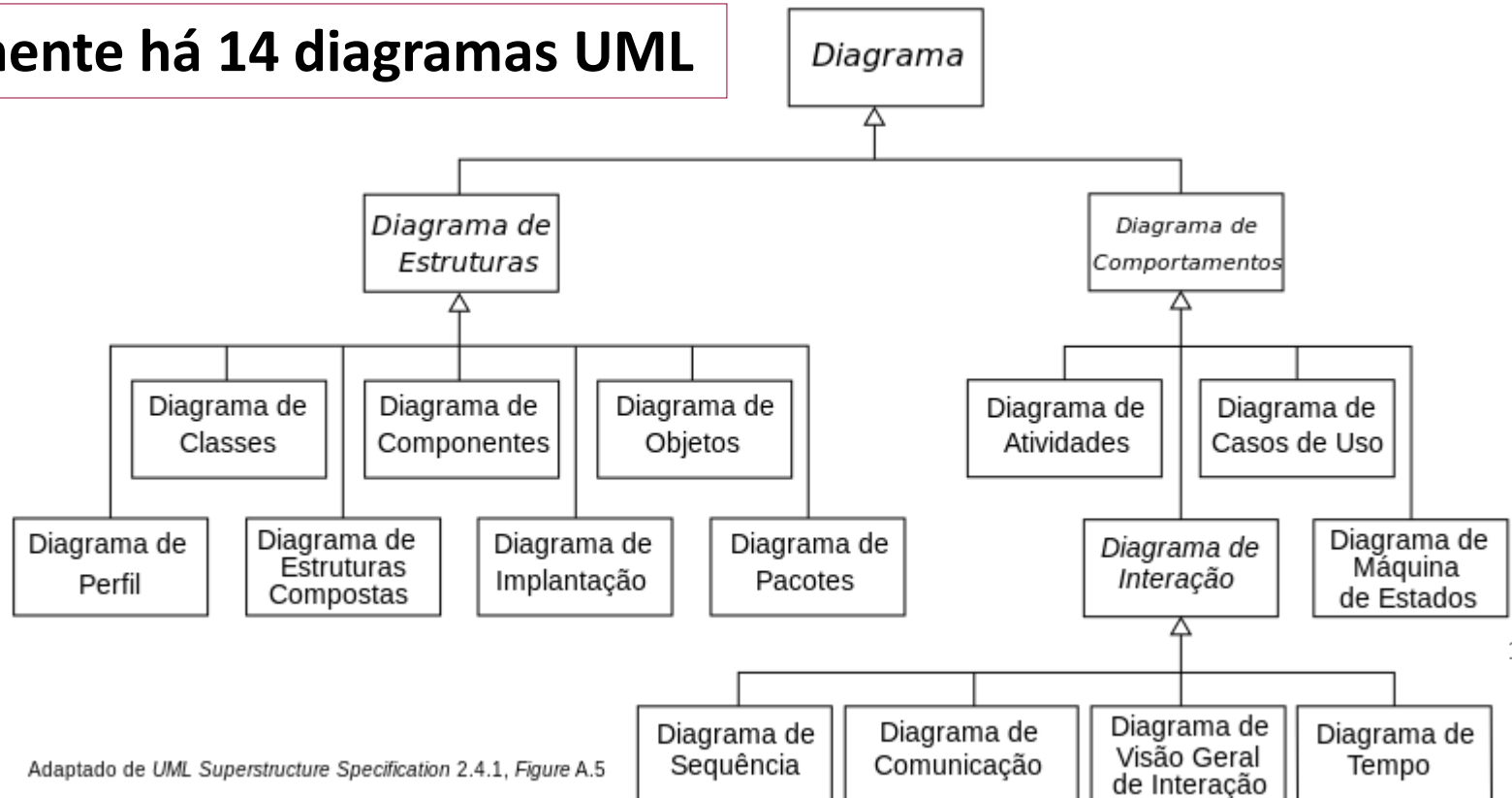
- Ajudar a **conceber as ideias**, em relação ao sistema que estivermos projetando;
- **Pensar antes de codificar**;
- **Apresentar as ideias ao grupo** de forma que todos possam interagir e discutir um determinado ponto;
- Aumentar a **participação e envolvimento do time**;
- **Documentar as ideias quando elas já estiverem bem consolidadas** para que novos integrantes e novos colaboradores possam acelerar sua compreensão dos sistemas desenvolvidos pelo grupo.

Diagramas UML

São representações gráficas de um conjunto de elementos. São **desenhados para visualizar um sistema de diferentes perspectivas**.



Atualmente há 14 diagramas UML



Adaptado de UML Superstructure Specification 2.4.1, Figure A.5

Diagramas UML

UML 2.0 em 2004 e UML 2.5 (versão atual) em 2015

UML 1.X	UML 2.x
Atividades	Atividades
Caso de Uso	Caso de Uso
Classe	Classe
Objetos	Objetos
Sequência	Sequência
Colaboração	Comunicação
Transição de Estados	Transição de Estados
	Pacotes
Componentes	Componentes
Implantação	Implantação
	Interatividade
	Tempo
	Perfil
	Estrutura Composta

UML – Mecanismos gerais

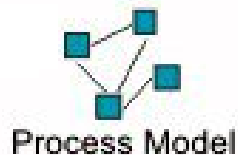
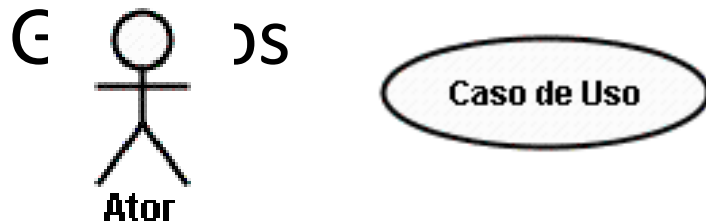
- Componentes da UML
 - Blocos de construção básicos
 - Regras que restringem como os blocos de construção podem ser associados
 - Mecanismos de uso geral
 - Estereótipos, Notas explicativas, Etiquetas valoradas, Restrições, Pacotes, OCL

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



- Estereótipos Textuais

`<<document>>`

`<<interface>>`

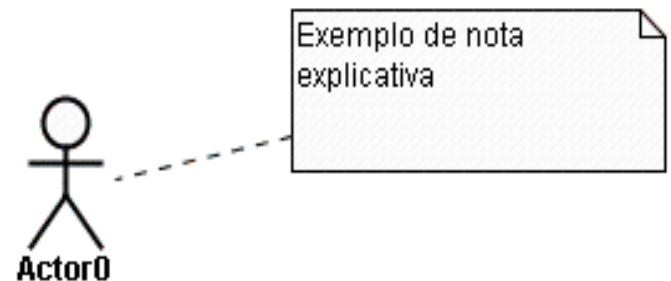
`<<entity>>`

`<<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}

<<boundary>>

Pedidos

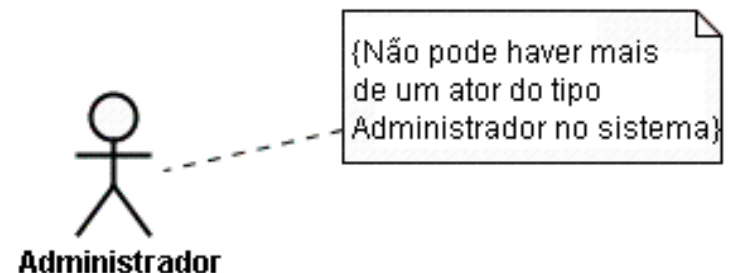
{autor=Maria Jocelia,
data de criação=17/09/07}

- numero : byte

+ Solicitar() : void

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

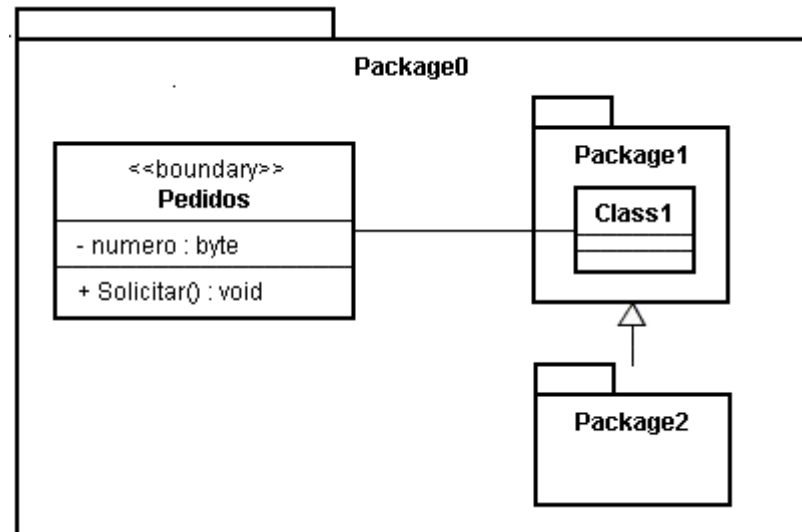
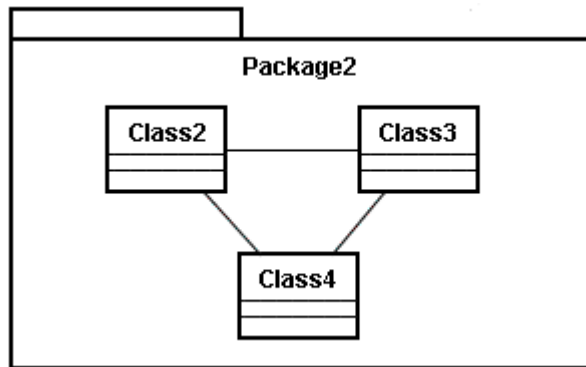


UML – Mecanismos gerais

- 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

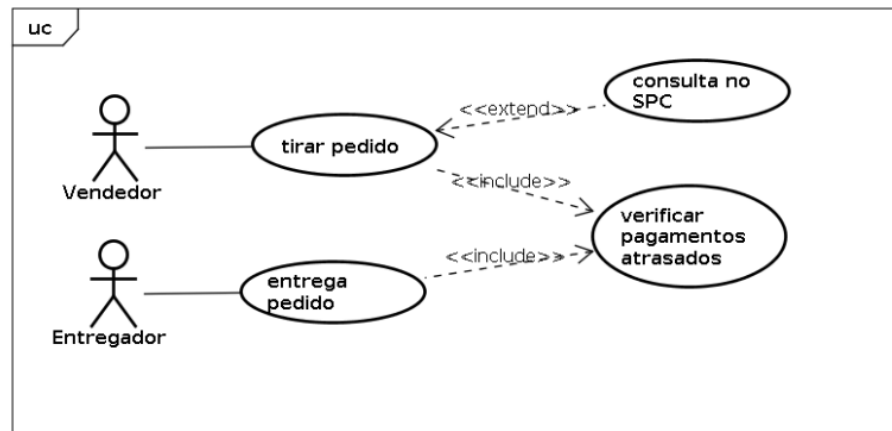
- OCL (Linguagem de restrição de objetos)
 - Linguagem formal para especificar restrições sobre diversos elementos em um modelo
 - Consiste de:
 - Contexto: Domínio no qual a declaração em OCL se aplica
 - Propriedade: um componente do contexto
 - Operação: O que deve ser aplicado sobre a propriedade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

Veiculo
- proprietario : Pessoa
- cor : String
- marca : String

Context Veículo

inv: self.proprietário.idade >= 18

Diagrama de Caso de Uso



Diagramas UML

1. Diagramas de Caso de Uso

- São especialmente importantes na **organização e modelagem** das principais funcionalidades de um sistema;
- Use Case (UC) é a especificação de **sequências de ações para atender a uma funcionalidade do sistema**, interagindo com seus agentes.

Elementos do diagrama:

- Atores
- Casos de uso
- Relacionamentos
 - Associação
 - Generalização
 - Dependência: Extensão e Inclusão
- Fronteira do sistema

Diagramas UML

1. Diagramas de Caso de Uso

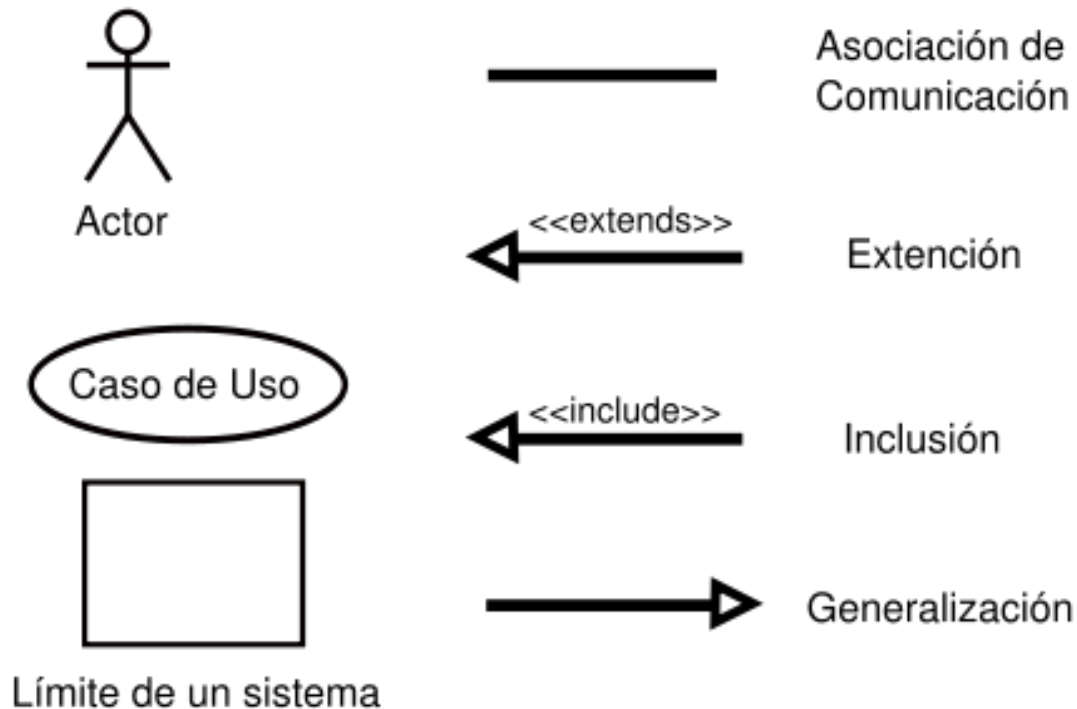
Devem responder (Jacobson):

- Quem são os atores?
- Quais são seus objetivos?
- Que pré-condições existem?
- Quais as tarefas principais realizadas?
- Que exceções devem ser consideradas?
- Que variações são possíveis nas interações?
- Que informações do sistema serão adquiridas, produzidas ou alteradas?

Diagramas UML

1. Diagramas de Caso de Uso

Opções disponíveis pelo diagrama:



Diagramas UML

1. Diagramas de Caso de Uso

- Servem para **facilitar o entendimento** de um sistema mostrando a sua “visão externa”;
- São usados para **modelar o contexto de um sistema**, subsistema ou classe;
- Uma das maneiras mais comuns de documentar os requisitos do sistema.
 - Delimitam o Sistema
 - Definem a funcionalidade do sistema



Diagramas UML

1. Diagramas de Caso de Uso

REPRESENTAÇÃO GRÁFICA

- A coleção dos use cases deverá especificar todas as formas existentes de uso do sistema.



Matricular aluno

**Solicitar
histórico**

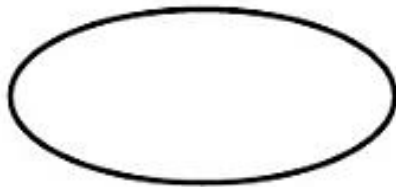
**Verificar
pré-requisitos**

Diagramas UML

1. Diagramas de Caso de Uso

REPRESENTAÇÃO GRÁFICA

- **Caso de Uso**



Função

- Serviço usado por um ou mais atores
- OU Seqüência de ações, executada pelo sistema, que gera um resultado

Ação: Procedimento computacional/algorítmico atômico

Diagramas UML

1. Diagramas de Caso de Uso

ATORES

- **Ator**



Emissor/Receptor

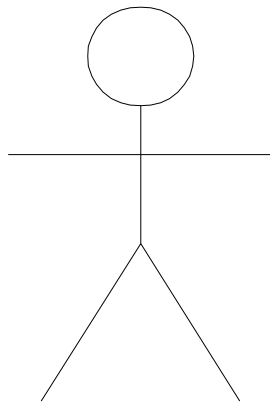
- Alguém ou alguma coisa (**fora do sistema**) que interage com o sistema
 - Estimula/solicita ações/eventos do sistema e recebe reações
- Agentes externos ao sistema

Um **Ator** pode ser um outro sistema que interage com o sistema que está sendo especificado

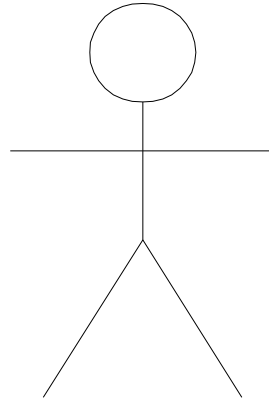
Diagramas UML

1. Diagramas de Caso de Uso

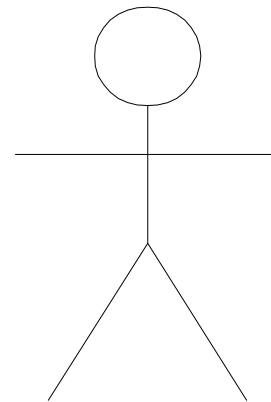
ATORES - Notação



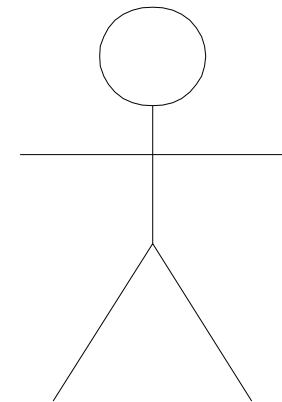
Professora



Secretária



**Sistema de controle
de pre-requisitos**



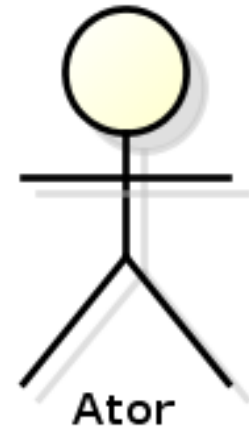
Estudante

Diagramas UML

1. Diagramas de Caso de Uso

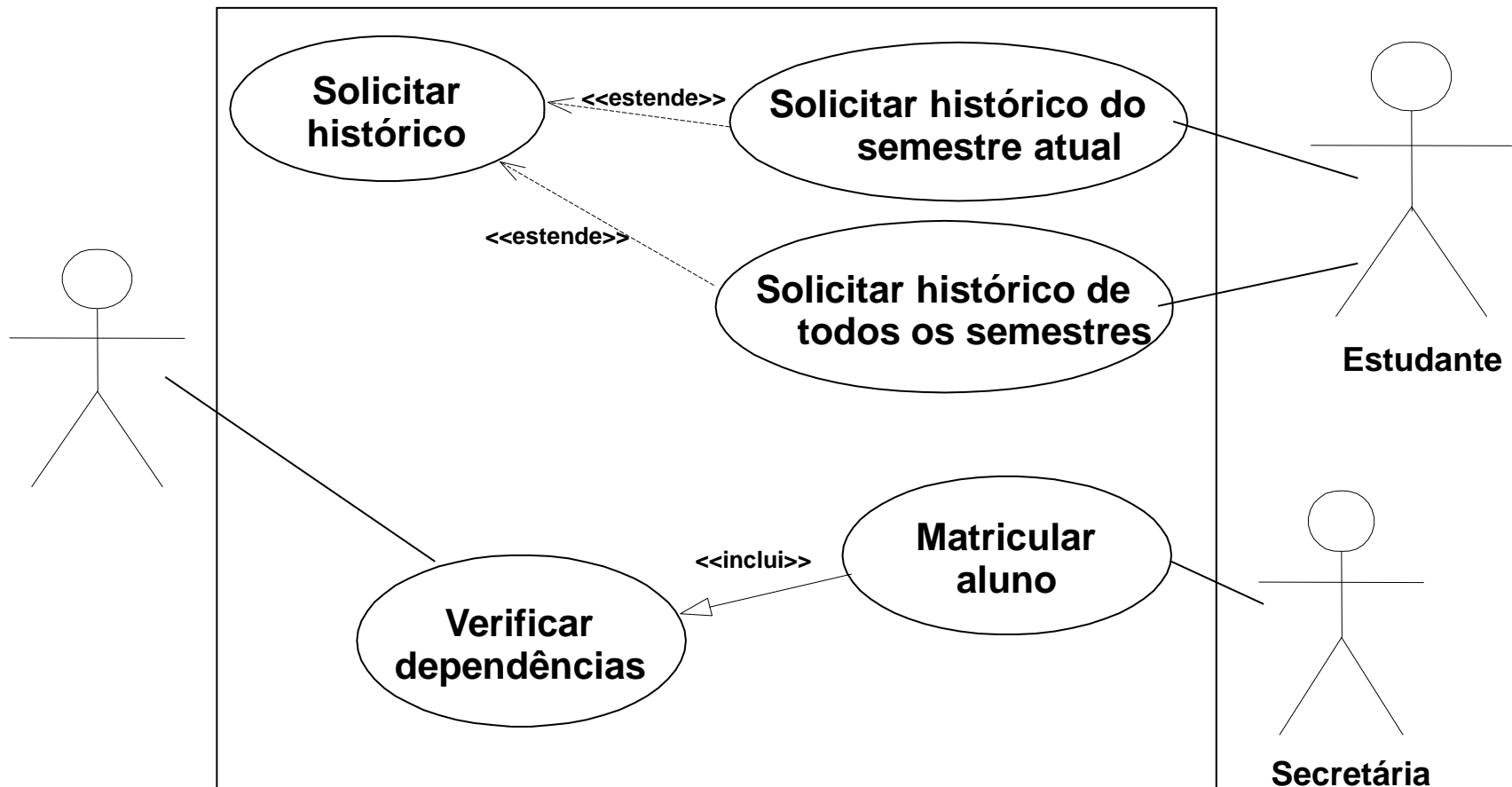
Atores: Exemplo

- **Cliente:** pessoa que compra produtos
- **Funcionário:** pessoa que realiza operações específicas na loja e atendimento ao cliente
- **Transportador:** empresa que entrega o produto
- **Sistema de pagamento:** sistema que debita valor da conta do cartão de crédito do cliente
- **Sistema de estoque:** sistema que controla o estoque de produtos na loja



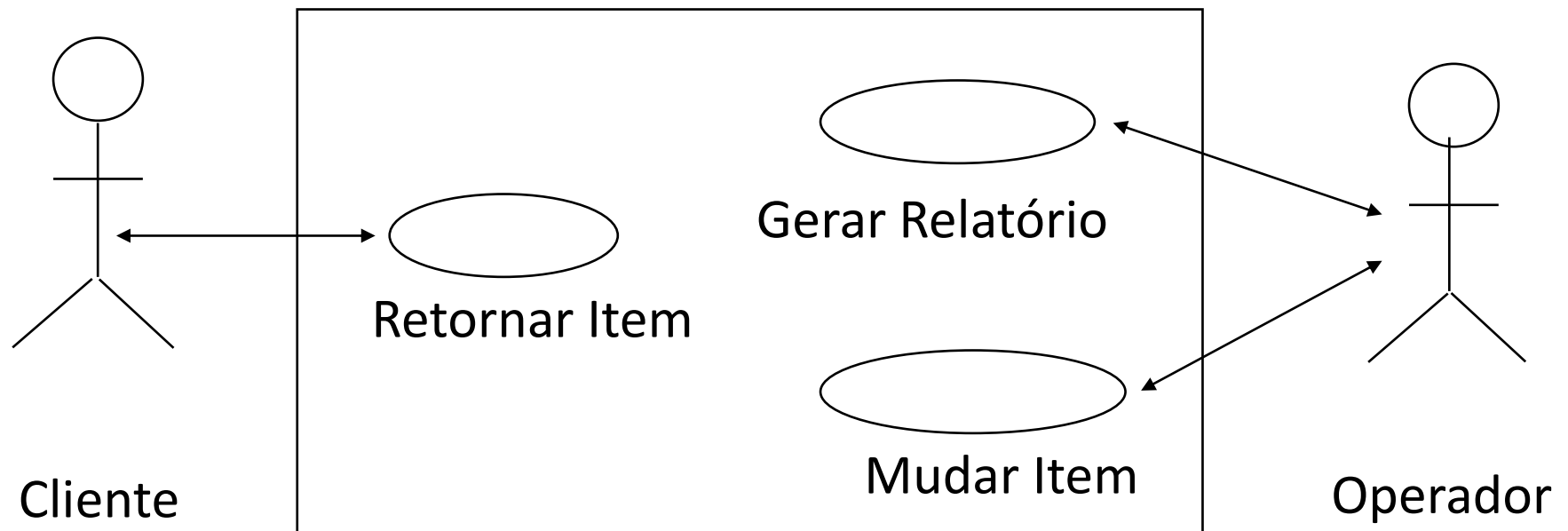
Diagramas UML

1. Diagramas de Caso de Uso



Diagramas UML

1. Diagramas de Caso de Uso



Diagramas UML

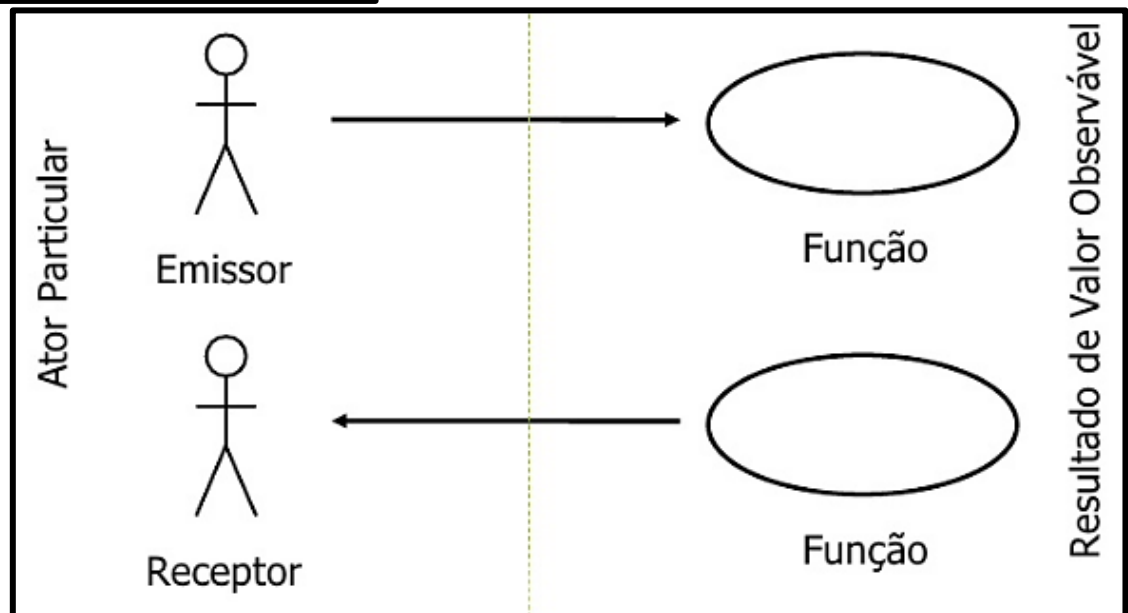
1. Diagramas de Caso de Uso

• Relacionamento / Associação



Relacionamento

- Relaciona atores com casos de uso
- Relaciona casos de uso



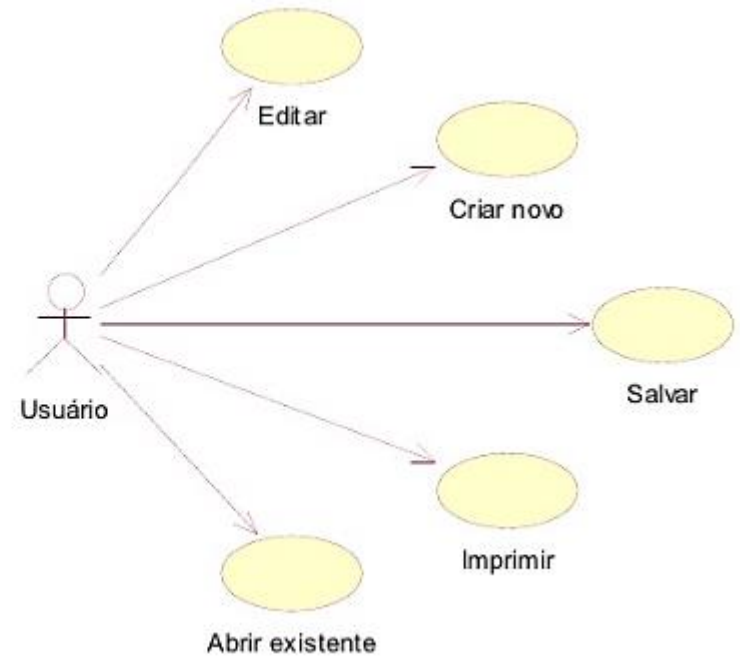
Diagramas UML

1. Diagramas de Caso de Uso

Caso de uso em alto nível para o Powerpoint



Casos de uso mais detalhados para o Powerpoint.

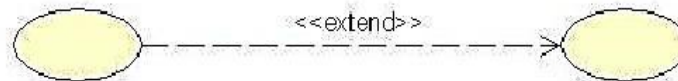
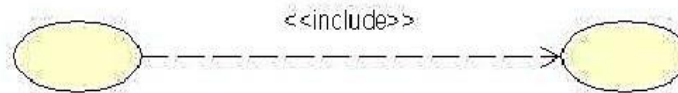


Diagramas UML

1. Diagramas de Caso de Uso

Há 3 possibilidades de relacionamento entre casos de uso:

**Inclusão, Extensão e
Generalização**



Diagramas UML

1. Diagramas de Caso de Uso

O **<<include>>** é usado quando casos compartilham comportamento comum com outros casos de uso (UC).

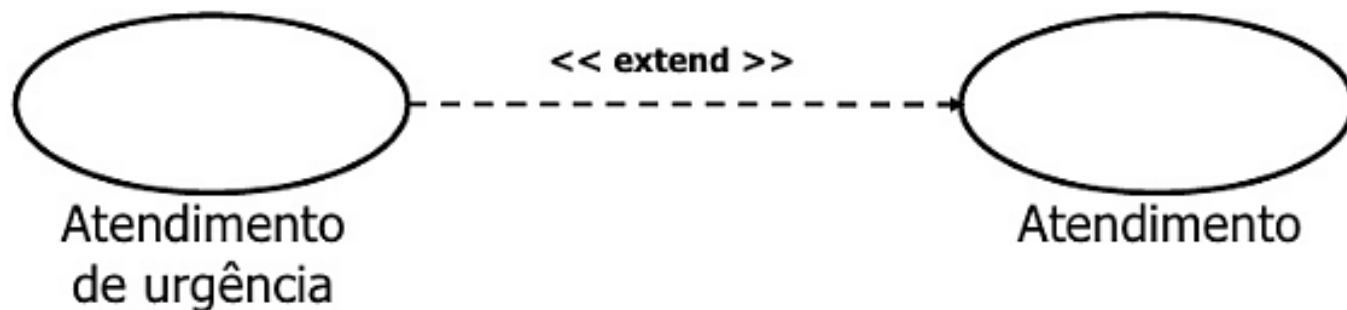
- Um caso de uso incorpora explicitamente o comportamento de outro;
- Funcionalidade comum é separada em um caso que é reutilizado por outros.



Diagramas UML

1. Diagramas de Caso de Uso

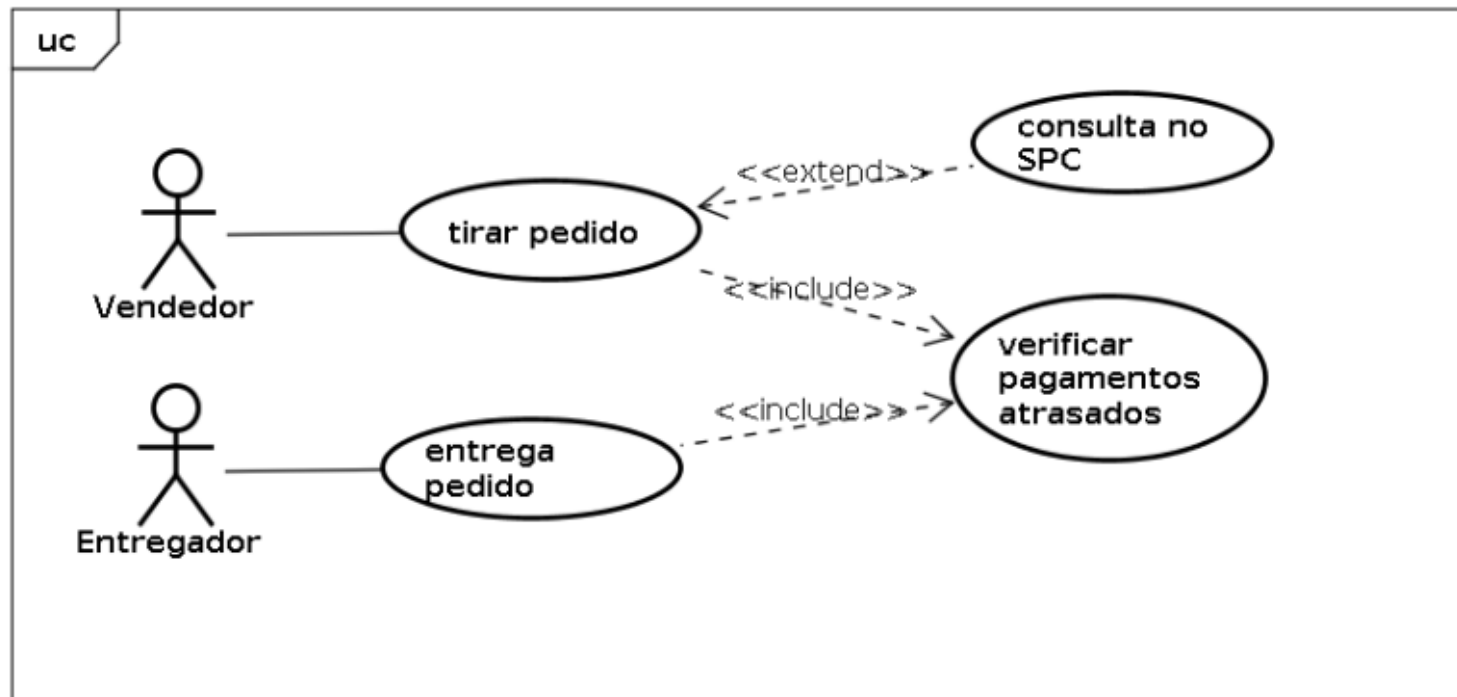
O **<<extend>>** é a utilização inversa da inclusão e pode alterar o UC que foi estendido. São subsequências que são executadas apenas em certos casos.



Quando o caso de uso **B** estende o caso de uso **A**, significa que quando o caso de uso **A** for executado o caso de uso **B** poderá (poderá – talvez não seja) ser executado também.

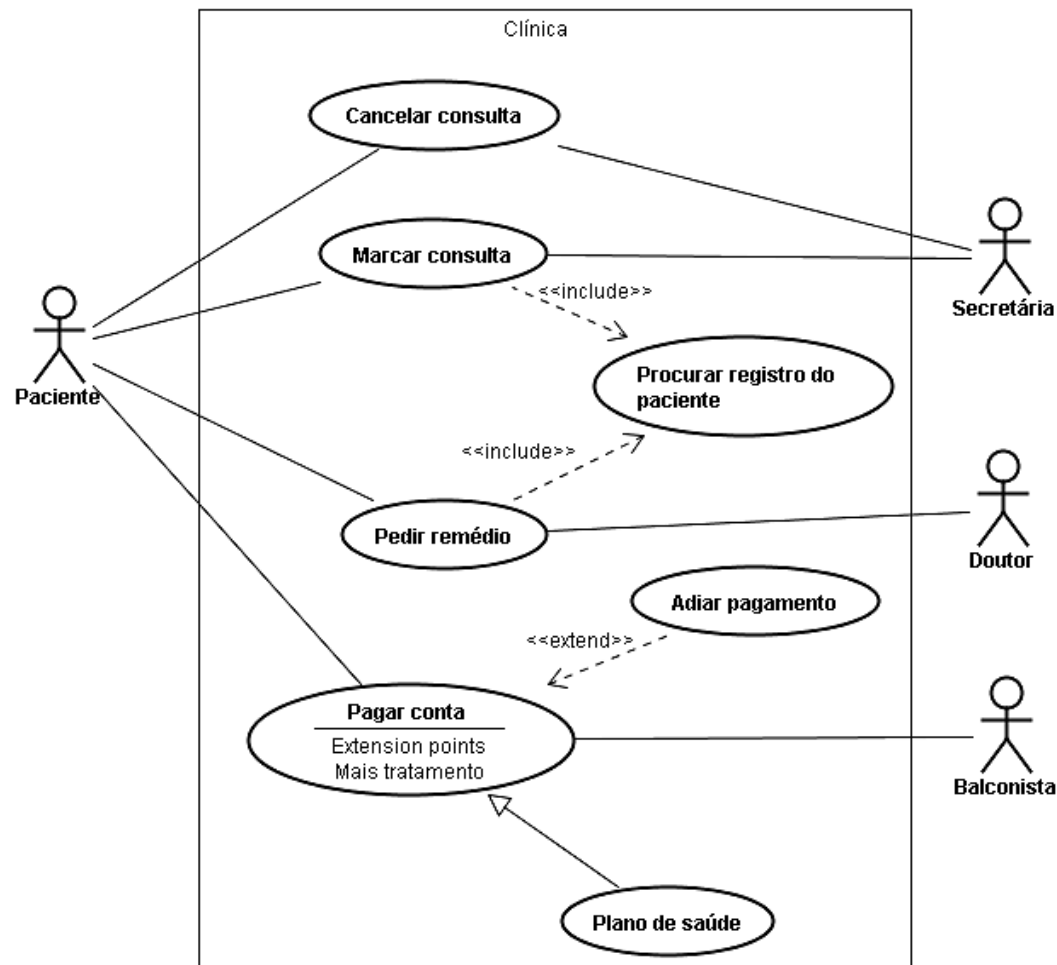
Diagramas UML

1. Diagramas de Caso de Uso



Diagramas UML

1. Diagramas de Caso de Uso

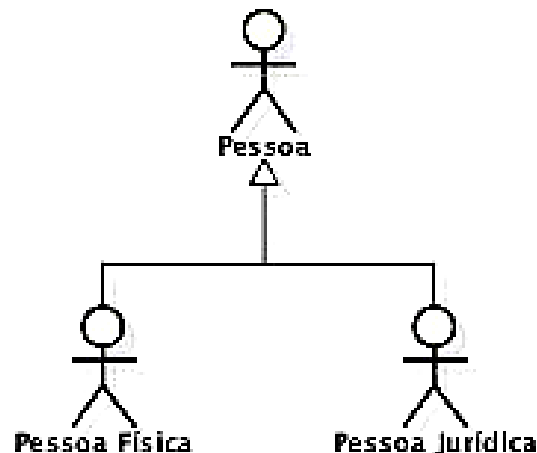


Diagramas UML

1. Diagramas de Caso de Uso

O <<generalização>>, ou <<especialização>> é quando um caso de uso filho herda o comportamento do caso de uso pai (permite modelar estruturas de aplicações em comum).

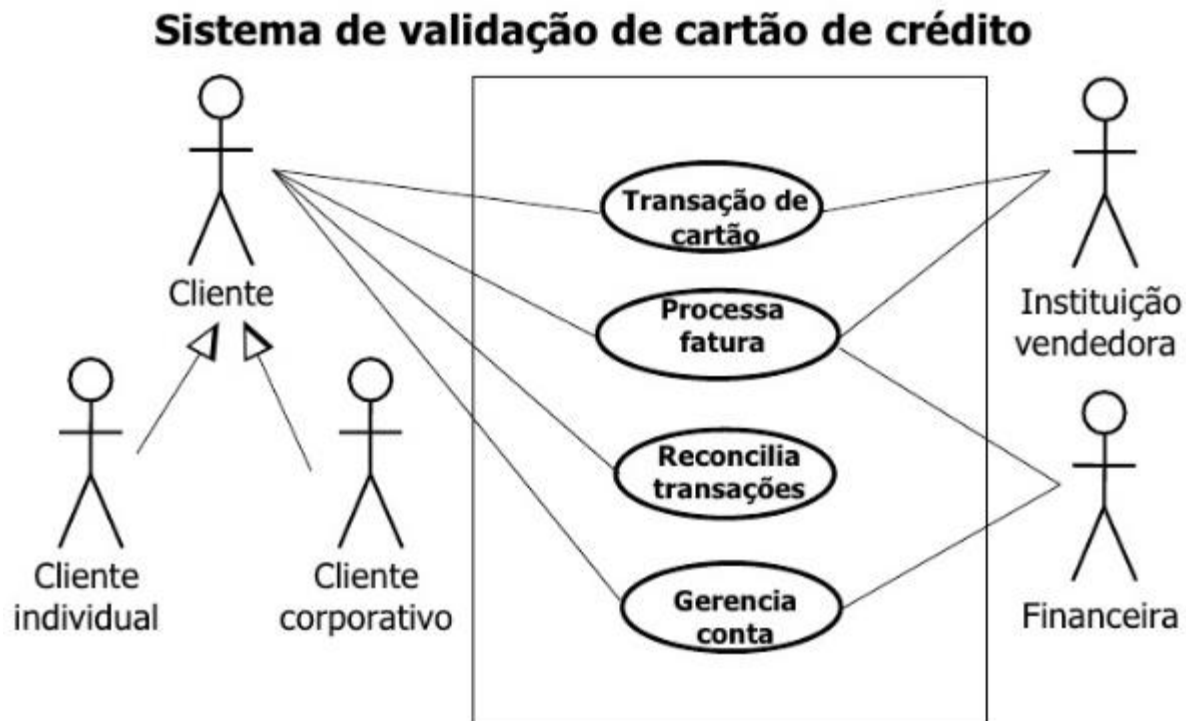
Quando o caso de uso **B** generaliza o caso de uso **C** isso significa que, além de fazer tudo que nele está especificado (ele = B), ele **também executará tudo que está especificado** no caso de uso **C**.



Diagramas UML

1. Diagramas de Caso de Uso

<<generalização>>



Diagramas UML

1. Diagramas de Caso de Uso

Descrição dos Casos de Uso

- O diagrama é insuficiente para dizer o que cada caso de uso faz;
- Deve-se descrever textualmente o fluxo de eventos de cada caso separadamente;
- Esta tarefa deve ser iniciada após alguma estabilidade dos casos de uso, para evitar perda de tempo.

Diagramas UML

1. Diagramas de Caso de Uso

Exemplo de Descrição de Casos de Uso

Um esboço inicial sobre “Sacar dinheiro” seria:

1. O use case inicia quando o Cliente insere um cartão no CA. Sistema lê e valida informação do cartão
2. Sistema pede a senha. Cliente entra com a senha. Sistema valida a senha.
3. Sistema pede seleção do serviço. Cliente escolhe “Sacar dinheiro”.
4. Sistema pede a quantia a sacar. Cliente informa.
5. Sistema pede seleção da conta (corrente, etc). Cliente informa.
6. Sistema comunica com a rede para validar a conta, senha e o valor a sacar.
7. Sistema comunica com a rede para validar a conta, senha e o valor a sacar.
8. Sistema entrega quantia solicitada.

Diagramas UML

1. Diagramas de Caso de Uso

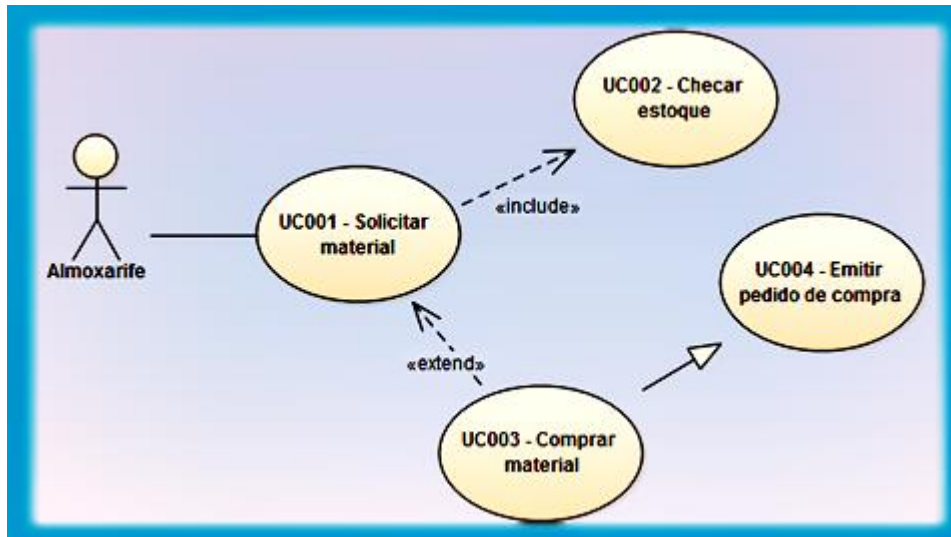
Ferramenta CASE - UML

- Uma ferramenta CASE (*Computer-Aided Software Engineering*) auxilia no desenho de diagramas de caso de uso;
- Há várias ferramentas disponíveis;
- Recomendamos o Jude UML:
- Download em **<http://jude.change-vision.com>**;
- Versão *Community* é gratuita.

JUDE
UML MODELING TOOL

Diagramas UML

1. Diagramas de Caso de Uso



GENERALIZATION = O UC “Comprar Material” generaliza o UC “Emitir pedido de compra”. Isso se dá porque em “Emitir pedido de compra” **existe** especificação de como se realiza o pedido de compra, **processo que não se dá somente no contexto do almoxarifado**, mas é o mesmo em qualquer área do negócio.

INCLUDE = O UC “Solicitar Material” faz include no UC “Checar Estoque”. Isso se dá porque **sempre** que houver a solicitação de material **sempre** haverá a consulta ao estoque para saber se o material está disponível. Se sempre haverá, o relacionamento correto é o include.

EXTEND = O caso de uso “Comprar Material” estende o caso de uso “Solicitar Material”. Isso se dá porque quando houver a solicitação de material, **caso o material não exista em estoque** (após consulta via o caso de uso “Checar estoque”) **poderá** ser solicitado a compra do item.

Ferramentas de apoio

Exemplos:

- Família Rational Rose gera código em Smalltalk, PowerBuilder, C++, J++ e VB:

<http://www-03.ibm.com/software/products/en/rosemod>

- ArgoUML

<http://argouml.tigris.org/>

- (lista de ferramentas que envolvem a UML), entre elas Jude (agora Astah) e Visual Paradigm

http://www.objectsbydesign.com/toolsumltools_byCompany.html

- StarUML

<http://staruml.sourceforge.net>

Amanhã (08/02) haverá atividade para
compor a 2ª nota

Lista de Exercícios (1,0) - online

Até a próxima aula...