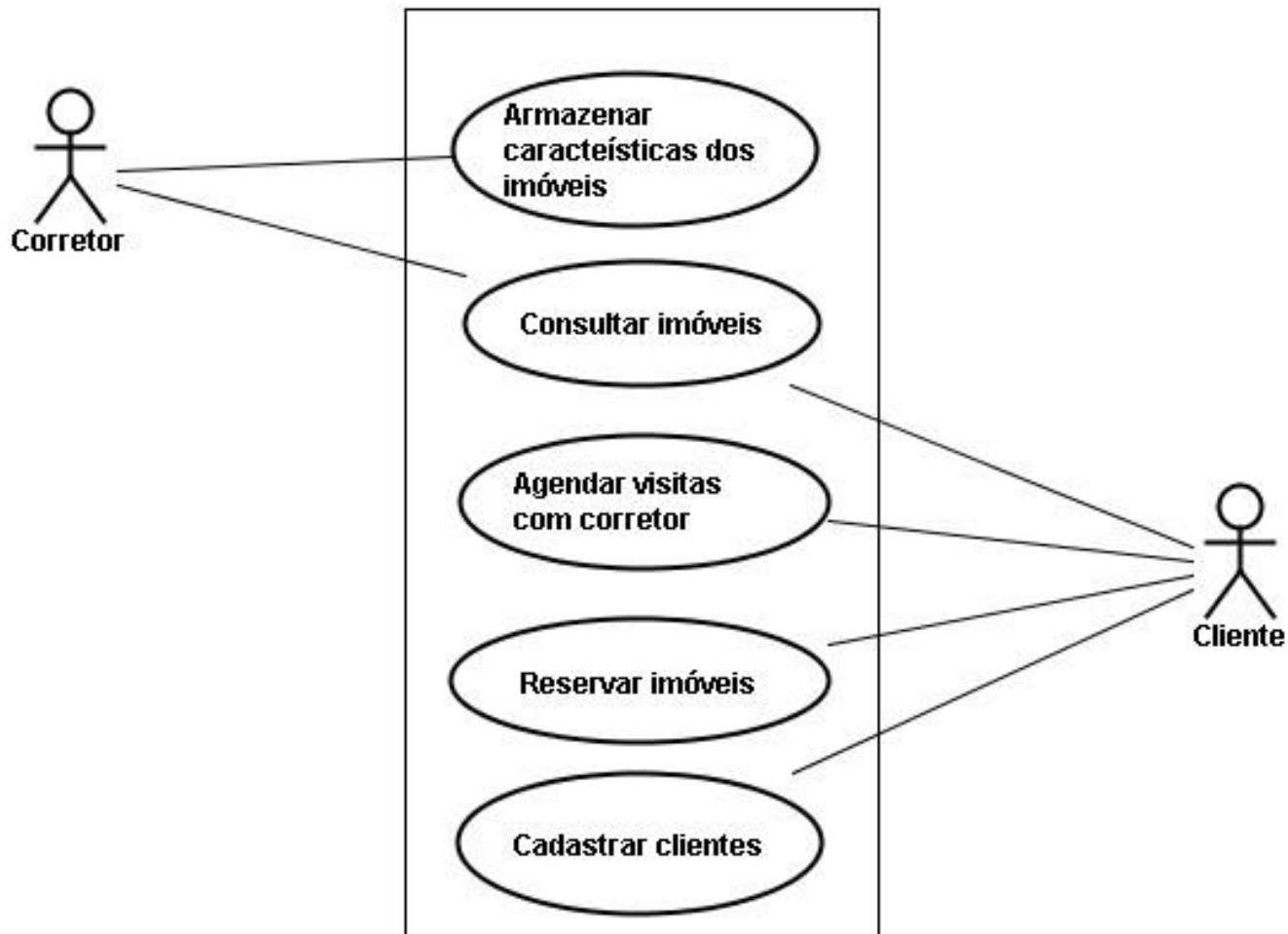


# Engenharia de Software

**Professor:**  
Zady Castaneda Salazar



# Aula 14 – UML: Casos de Uso



# UML Unified Modeling Language

---

Na área de Engenharia de Software, a Linguagem de Modelagem Unificada (do inglês, **UML - Unified Modeling Language**) é uma **linguagem de modelagem** que permite representar um sistema de forma padronizada.

# UML Unified Modeling Language

---

**UML** é uma linguagem para visualização, especificação, construção e documentação de artefatos de um software orientado a objeto.



# UML Unified Modeling Language

---

A UML (Unified Modeling Language) **não é uma metodologia de desenvolvimento**, o que significa que ela não diz para você o que fazer primeiro e/ou como projetar seu sistema, mas ela lhe **auxilia a visualizar seu desenho e a comunicação entre os objetos**(e em certos casos a identificação dos processos).

# UML Unified Modeling Language

---

## Vantagem:

Permite que desenvolvedores visualizem os produtos de seus trabalhos em diagramas padronizados.

Qualquer sistema, seja qual for o tipo, modelado corretamente, será:

- consistente,
- fácil de se comunicar com outras aplicações,
- simples de ser atualizado,
- compreensível.

# UML Unified Modeling Language

---

Por isso e muitas outras razões, o bom entendimento da UML não é apenas aprender a simbologia e o seu significado, mas também significa aprender a modelar orientado a objetos no estado da arte.

A UML é muito mais que a padronização de uma notação. É também o desenvolvimento de novos conceitos não normalmente usados.

# UML Unified Modeling Language

---

É única para todas as atividades: análise de requisitos, análise, design (projeto), programação e testes.

## Análise de requisitos

Esta fase captura as intenções e necessidades dos usuários do sistema a ser desenvolvido através do uso de funções chamadas “**casos de uso**”.

O diagrama de “**casos de uso**” mostrará o que os atores externos, ou seja, os usuários do futuro sistema deverão esperar do aplicativo, conhecendo toda sua funcionalidade sem importar como esta será implementada.





# UML Unified Modeling Language

---

É única para todas as atividades: análise de requisitos, análise, design (projeto), programação e testes.



## Análise

Está preocupada com as primeiras abstrações (classes e objetos) e mecanismos que estarão presentes no domínio do problema.

As classes são modeladas e ligadas através de relacionamentos com outras classes, e são descritas no **Diagrama de Classe**.

Na análise, só serão modeladas classes que pertençam ao domínio principal do problema do software, ou seja, classes técnicas que gerenciem banco de dados, interface, comunicação, concorrência e outros não estarão presentes neste diagrama.

# UML Unified Modeling Language

---

É única para todas as atividades: análise de requisitos, análise, design (projeto), programação e testes.



## Design(projeto)

Na fase de design, o resultado da análise é expandido em soluções técnicas.

Novas classes serão adicionadas para prover uma infra-estrutura técnica: a interface do usuário e de periféricos, gerenciamento de banco de dados, comunicação com outros sistemas, dentre outros.

As classes do domínio do problema modeladas na fase de análise são mescladas nessa nova infra-estrutura técnica tornando possível alterar tanto o domínio do problema quanto a infra-estrutura.

O design resulta no detalhamento das especificações para a fase de programação do sistema.

# UML Unified Modeling Language

---

É única para todas as atividades: análise de requisitos, análise, design (projeto), programação e testes.



## Codificação(programação)

Na fase de programação, as classes provenientes do design são convertidas para o código da linguagem orientada a objetos escolhida.(C# e Java)

Dependendo da capacidade da linguagem usada, essa conversão pode ser uma tarefa fácil ou muito complicada.

A programação é uma fase separada e distinta onde os modelos criados são convertidos em código.

# UML Unified Modeling Language

---

É única para todas as atividades: análise de requisitos, análise, design (projeto), programação e testes.



## Teste

Um sistema normalmente é rodado em testes de unidade, integração, e aceitação.

Os testes de unidade são para classes individuais ou grupos de classes e são geralmente testados pelo programador.

Os testes de integração são aplicados já usando as classes e componentes integrados para se confirmar se as classes estão cooperando uma com as outras como especificado nos modelos.

Os testes de aceitação observam o sistema como uma "caixa preta" e verificam se o sistema está funcionando como o especificado nos primeiros diagramas de "caso de uso".

O sistema será testado pelo usuário final e verificará se os resultados mostrados estão realmente de acordo com as intenções do usuário final.

# UML Unified Modeling Language

---

É única para todas as atividades: análise de requisitos, análise, design (projeto), programação e testes.



## Implantação

Os diagramas de implementação são utilizados para representar a arquitetura física de um sistema.

Os modelos construídos a partir desses diagramas são chamados diagramas de implementação.

Há dois tipos de diagramas de implementação:

- Diagramas de componentes
- Diagramas de implantação

# UML Unified Modeling Language

---

## Versões: Historia

VERSION	ADOPTION DATE	URL
2.5.1	December 2017	<a href="https://www.omg.org/spec/UML/2.5.1">https://www.omg.org/spec/UML/2.5.1</a>
2.4.1	July 2011	<a href="https://www.omg.org/spec/UML/2.4.1">https://www.omg.org/spec/UML/2.4.1</a>
2.3	May 2010	<a href="https://www.omg.org/spec/UML/2.3">https://www.omg.org/spec/UML/2.3</a>
2.2	January 2009	<a href="https://www.omg.org/spec/UML/2.2">https://www.omg.org/spec/UML/2.2</a>
2.1.2	October 2007	<a href="https://www.omg.org/spec/UML/2.1.2">https://www.omg.org/spec/UML/2.1.2</a>
2.0	July 2005	<a href="https://www.omg.org/spec/UML/2.0">https://www.omg.org/spec/UML/2.0</a>
1.5	March 2003	<a href="https://www.omg.org/spec/UML/1.5">https://www.omg.org/spec/UML/1.5</a>
1.4	September 2001	<a href="https://www.omg.org/spec/UML/1.4">https://www.omg.org/spec/UML/1.4</a>
1.3	February 2000	<a href="https://www.omg.org/spec/UML/1.3">https://www.omg.org/spec/UML/1.3</a>
1.2	July 1999	<a href="https://www.omg.org/spec/UML/1.2">https://www.omg.org/spec/UML/1.2</a>
1.1	December 1997	<a href="https://www.omg.org/spec/UML/1.1">https://www.omg.org/spec/UML/1.1</a>

Fonte: <https://www.omg.org/spec/UML/>

# UML Unified Modeling Language

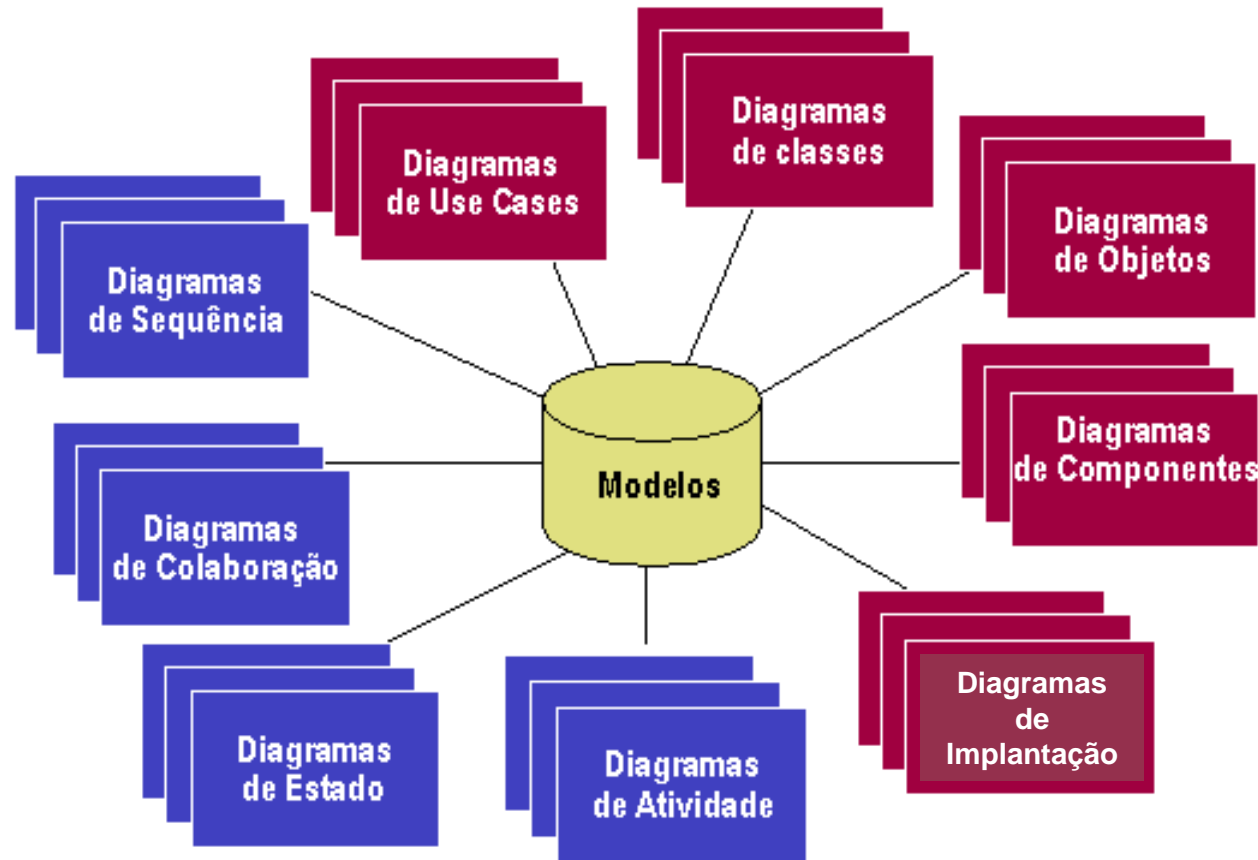
---

## Tipos dos Diagramas UML

- Diagrama estáticos
  - Diagrama de casos de uso
  - Diagrama de classes
  - Diagrama de objetos
  - Diagrama de componentes
  - Diagrama de implantação
- Diagrama dinâmicos
  - Diagrama de sequência
  - Diagrama de colaborações
  - Diagrama de estados
  - Diagrama de atividades

# UML Unified Modeling Language

O modelo do sistema é representado pelos dois conjuntos de diagrama, estático e dinâmico(ver figura abaixo).



*Fig1: Vermelho: estático (ou estrutural); Azul: dinâmico (ou comportamental).*



# Casos de Uso

---

- Um software frequentemente é um produto complexo, e sua descrição envolve a identificação e documentação de vários casos de uso, cada um deles descrevendo uma "fatia" do que o software ou uma de suas partes deverá oferecer.
- É importante notar que **não descreve como o software deverá ser construído**, mas sim como ele deverá se comportar quando estiver pronto.
- Normalmente evitam o uso de termos técnicos, preferindo a linguagem do utilizador final, são empregados tanto por quem desenvolve o software quanto pelos utilizadores do software.
-

# Casos de Uso

---

- Um caso de uso pode "**incluir**" outra funcionalidade de caso de uso ou "**estender**" outro caso de uso com seu próprio comportamento.
- Casos de uso são tipicamente relacionados a "atores".
- **Um ator é um humano, entidade ou dispositivos** que interagem com o sistema para executar um trabalho.

# Casos de Uso

---

- Casos de uso são narrativas em texto, descrevendo a unidade funcional, e são amplamente utilizados para descobrir e registrar os requisitos de sistemas.
- Cada caso de uso tem uma descrição o qual descreve a funcionalidade que irá ser construída no sistema proposto.

# Especificação de Casos de Uso

---

Um modelo padrão de Caso de Uso (use case) deve conter:

- ✓ Público Alvo
- ✓ Objetivo do documento
- ✓ Definições e Abreviaturas
- ✓ Fluxo Normal
- ✓ Cenário
- ✓ Fluxo Alternativos
- ✓ Regras de Negócio
- ✓ Diagramas de Casos de Uso
- ✓ Histórico de alterações deste documento

# Especificação de Casos de Uso

---

- Como representar os de Casos de Uso ?
  - Identificação do Caso de Uso
  - Nome do Caso de Uso
  - Ator: ator que interage com o caso de uso
  - Pré-condições: o estado do sistema para que o caso de uso possa iniciar
  - Pós-condições: o estado do sistema após a execução do caso de uso
  - Sequência de Eventos
  - Requisitos Não-Funcionais

# Especificação de Casos de Uso

---

- Como representar os de Casos de Uso ?

- Sequência de Eventos

## Sequência Típica de Eventos

### Ação do Ator

Ações numeradas de ator

### Resposta do Sistema

Descrição numerada  
das respostas do  
sistema

## Sequências Alternativas

Alternativas que podem surgir  
por número de linha:  
descrição de exceções.

# Especificação de Casos de Uso

---

## Exemplo:

- Identificação: UC1
- Caso de uso: Sacar dinheiro no caixa eletrônico
- Ator: Cliente
- Pré-Condições: o Cliente possui cartão do banco e senha cadastrada.
- Pós-Condições: lançada a transação na conta do Cliente, atualizado o saldo da conta corrente e liberado o dinheiro.
- Sequência Típica de Eventos (Fluxo Básico):

Ação do ator	Resposta do sistema
1. Este caso de uso começa quando o Cliente realiza a leitura do cartão do banco no caixa eletrônico	
2. O Cliente informa a sua senha	3. O sistema valida a conta corrente e senha do Cliente, autorizando a operação
4. O Cliente informa o valor do saque	5. O sistema autoriza o saque e lança o débito na conta corrente do Cliente
	6. O sistema libera o dinheiro

# Especificação de Casos de Uso

---

## Exemplo:

### Sequências Alternativas (Fluxos Alternativos):

#### 3a. Cliente Inválido:

1. O sistema não reconhece a conta corrente e senha do Cliente como válida
2. A operação é cancelada

#### 5a: Fundos Insuficientes:

1. O sistema não autoriza o valor solicitado para saque pelo Cliente
2. A operação é cancelada

### Requisitos Não-Funcionais

- Resposta do sistema deve ocorrer em no máximo 30 seg em 90 % dos casos



# Especificação e Modelagem de Casos de Uso

---

- **Etapas para Especificação de Casos de Uso:**
  1. Identificar as **requisições** de **entrada** e **saída** para usuários
  2. Identificar as **requisições** envolvendo **outros sistemas** existentes
  3. Descrever as **funções** (casos de uso) – funcionalidades
  4. Identificação de **atores** e **interação** (relacionamento)
  5. Identificar as **restrições**
  6. Especificação de **cenários**
  7. Elaboração de **caso de uso** descrito textualmente
  8. Construção de **diagramas** mostrando as dependências funcionais

# Diagrama de Casos de Uso

---

- A modelagem de um diagrama de caso de uso é uma técnica usada para descrever e definir os requisitos funcionais de um sistema.

## **Objetivo**

Auxiliar a comunicação entre os analistas e o cliente.

# Diagrama de Casos de Uso

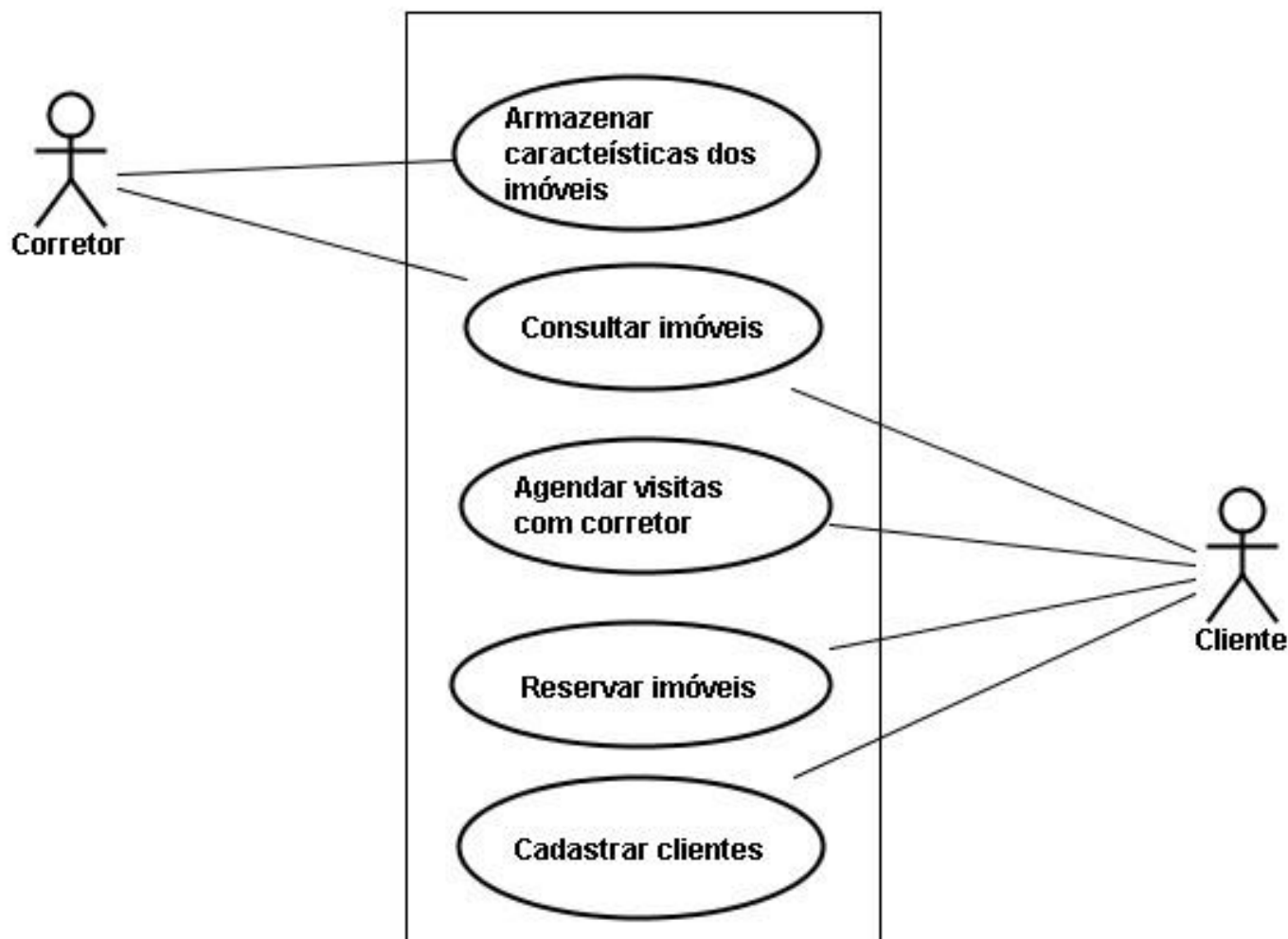
---

Um diagrama de Caso de Uso descreve um cenário que **mostra as funcionalidades** do sistema do **ponto de vista do usuário**.

O cliente deve ver no diagrama de Casos de Uso as **principais funcionalidades** de seu sistema.

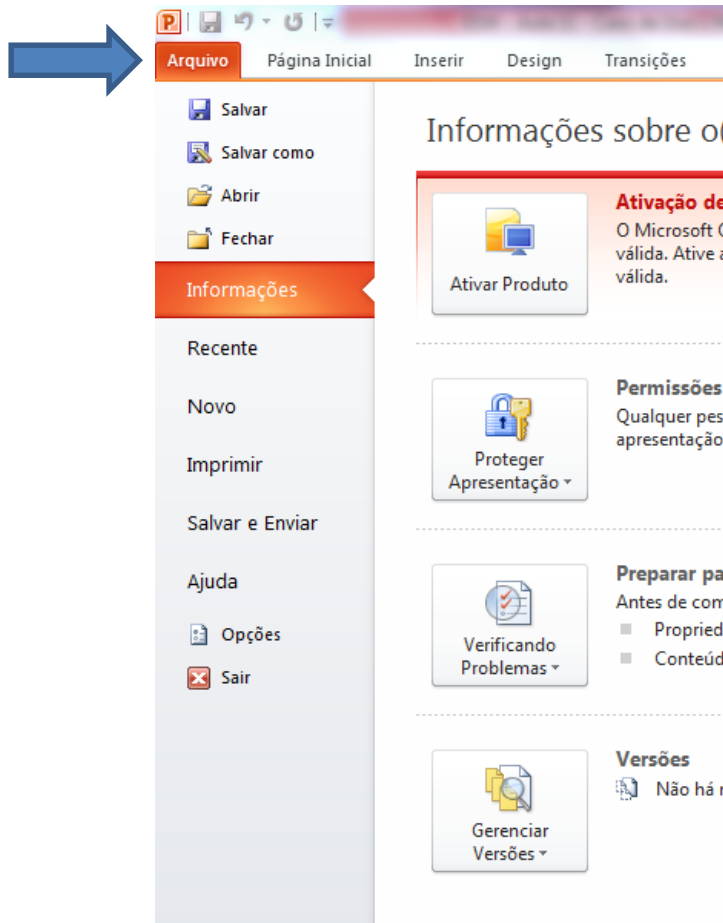
# Diagrama de Casos de Uso

## Exemplo de diagrama de Casos de Uso

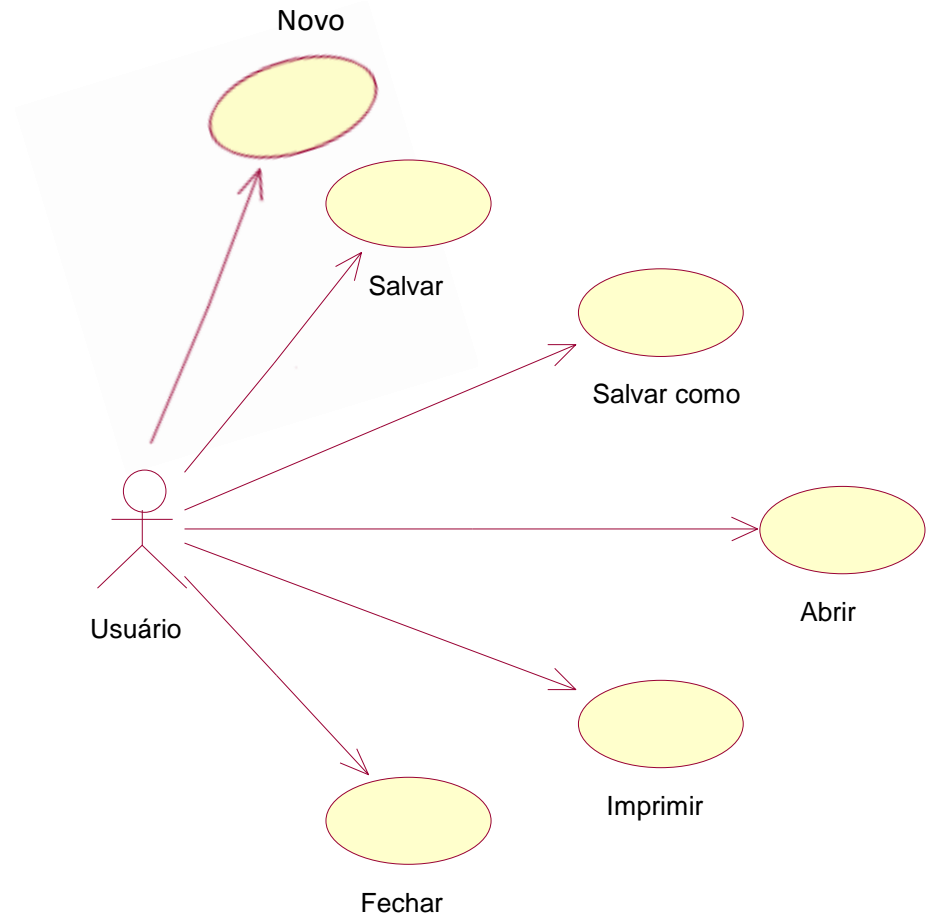


# Diagrama de Casos de Uso

## Exemplo: Diagrama de Casos de uso



## Menu **ARQUIVO** do Powerpoint



# Modelagem de Casos de Uso

---

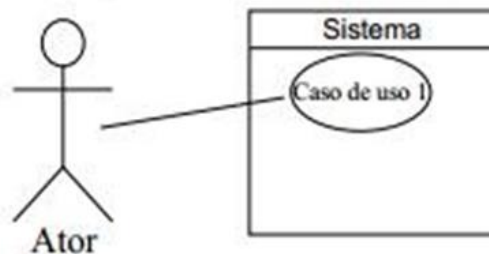
- **Elementos do diagrama de Casos de Uso**

**Ator:** representa o papel de uma entidade externa ao sistema como um usuário, um hardware, ou outro sistema que interage com o sistema modelado. Os atores iniciam a comunicação com o sistema através dos casos de uso.

**Caso de uso:** representa uma sequência de ações executadas pelo sistema e um conjunto de serviço que o sistema fornece aos usuários.

**Relacionamento:** união de um ou mais elementos através de associações, e tanto atores quanto casos de uso podem possuir relacionamentos

**Sistema:** projeto ou um modulo do projeto.

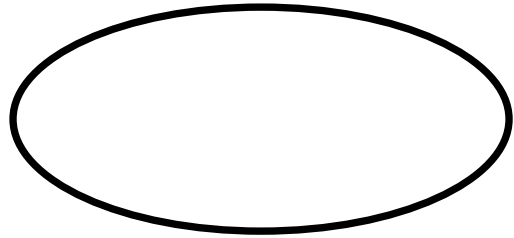


## Exemplo de Atores

- **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

## Representação de Casos de Usos

- **Caso de Uso**



Função

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



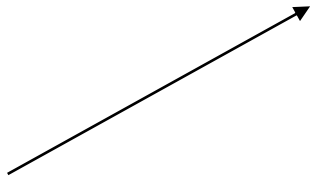
# Modelagem de Casos de Uso

---

- **Tipos de Relacionamento:**
  - Associação
  - Inclusão (*include*)
  - Extensão (*extend*)
  - Generalização/Especialização (*Generalization*)

# Representação de Associações de Casos de Usos

- **Relacionamento / Associação**



- Relaciona atores com casos de uso

Relacionamento

## Associação

- Uma **associação entre um ator e um caso de uso** significa que estímulos podem ser enviados entre atores e casos de uso, que se comunicam entre si
- Provêem uma visão geral das funcionalidades do sistema.

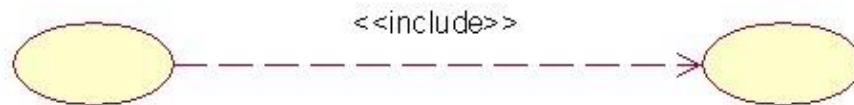
## Associação

- Demonstra que o Ator utiliza a função do sistema representada pelo Caso de Uso.
  - ✓ Requisitando a execução da função
  - ✓ Recebendo o resultado produzido pela função
- Representada por uma reta ligando o Ator ao Caso de Uso
  - ✓ Direcionada ou não

## Associações entre Casos de Usos

- **Inclusão:**

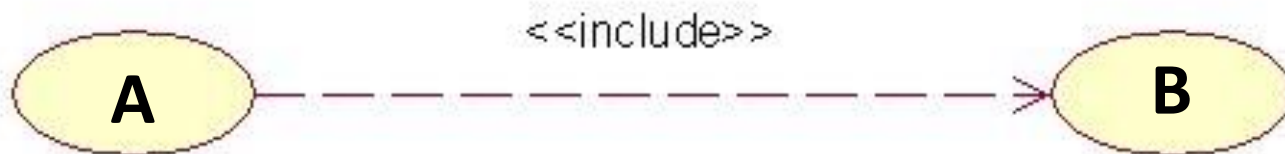
- ✓ Utilizado quando um caso de uso é usado dentro de outro caso de uso.
- ✓ Os relacionamentos de inclusão indicam obrigatoriedade.
- ✓ A execução do primeiro obriga a execução do segundo.



## Associações entre Casos de Usos

- **Inclusão:**

Quando o caso de uso **A** “inclui” o caso de uso **B**, significa que **sempre** que o caso de uso **A** for executado o caso de uso **B** também será executado. A direção do relacionamento é do caso de uso que está **incluindo** para o caso de uso **incluído**.

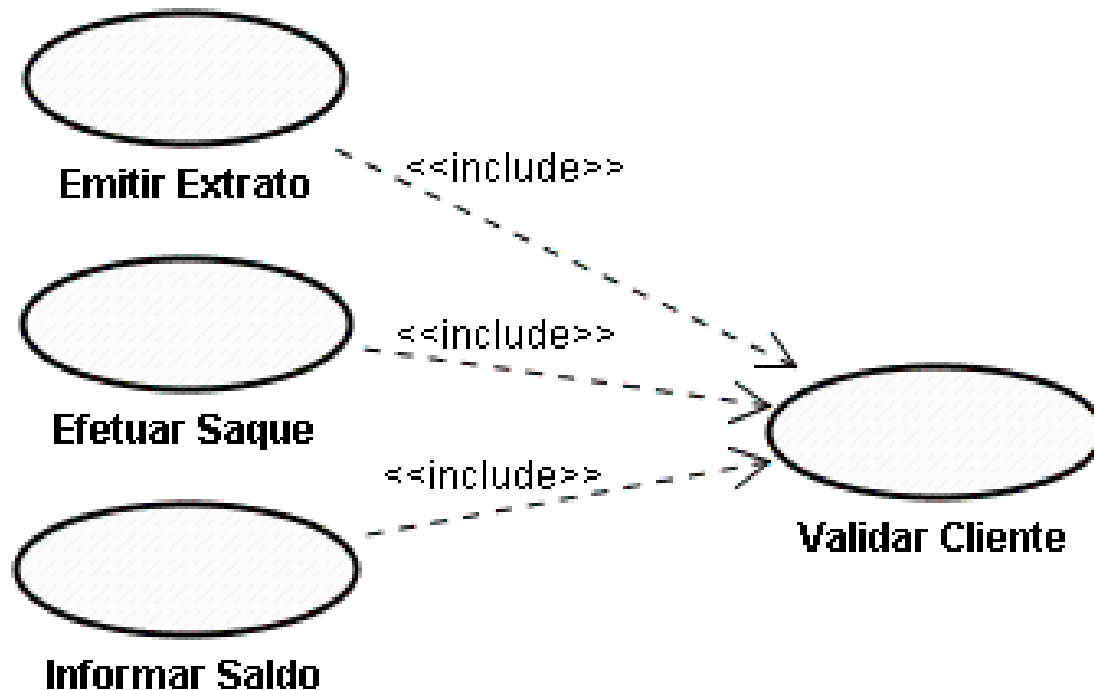


# Modelagem de Casos de Uso

---

## Exemplo de Inclusão

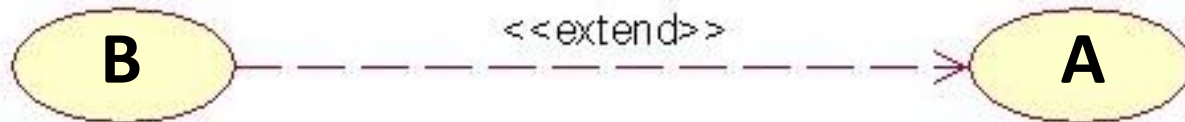
Para “**Emitir Extrato**”, “**Efetuar Saque**” e “**Informar Saldo**” necessitam da senha, então cria-se um novo caso de uso “**Validar cliente**” que inclui essa funcionalidade e permite validar qual é o cliente que va a realizar a operação.



## Associações entre Casos de Usos

- **Extensão:**

- ✓ Utilizar quando se tem dois casos de usos que fazem algo parecido, só que o caso de uso **B** faz alguma coisa a mais que **A**. **B** estende **A**.
- ✓ **B** representa alguma **situação não muito comum** que ocorre em **A** mediante a **satisfação de uma pré-condição**.

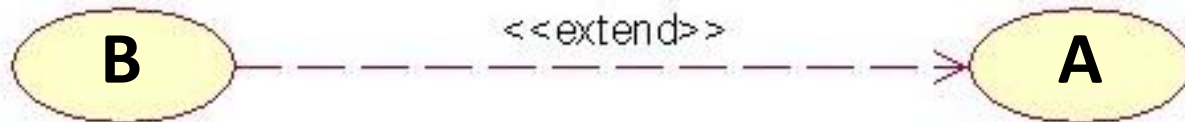




## Associações entre Casos de Usos

- **Extensão:**

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. A direção do relacionamento é do caso de uso **extensor**(aqui o caso de uso **B**) para o caso de uso **estendido** (aqui o caso de uso **A**).

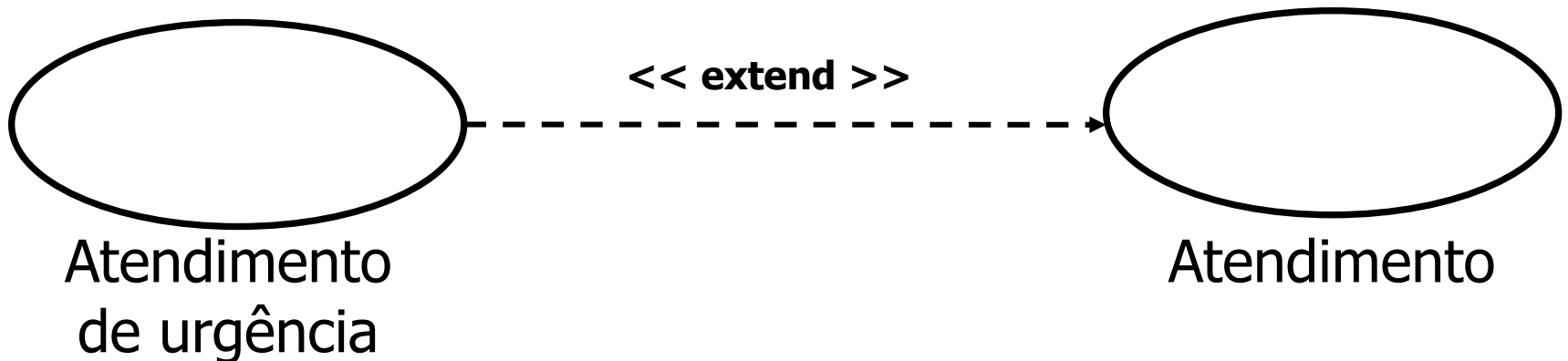


# Modelagem de Casos de Uso

---

## Exemplo de Extensão

No caso de uso **Atendimento de urgência** representa uma situação não muito comum que ocorre no caso de uso **Atendimento** quando se cumpre a pré-condição por exemplo, da chegada de um paciente vitima de um acidente de transito.



# Associações entre Casos de Usos

- **Generalização/Especialização (*Generalization*):**
  - ✓ Um relacionamento entre um caso de uso genérico para um mais específico, que herda todas as características de seu pai.



## Associações entre Casos de Usos

- **Generalização/Especialização (*Generalization*):**  
Quando o caso de uso **B** generaliza o caso de uso **C** isso significa que, além de fazer tudo que nele está especificado, **B** também **executará tudo que está especificado** no caso de uso **C**.

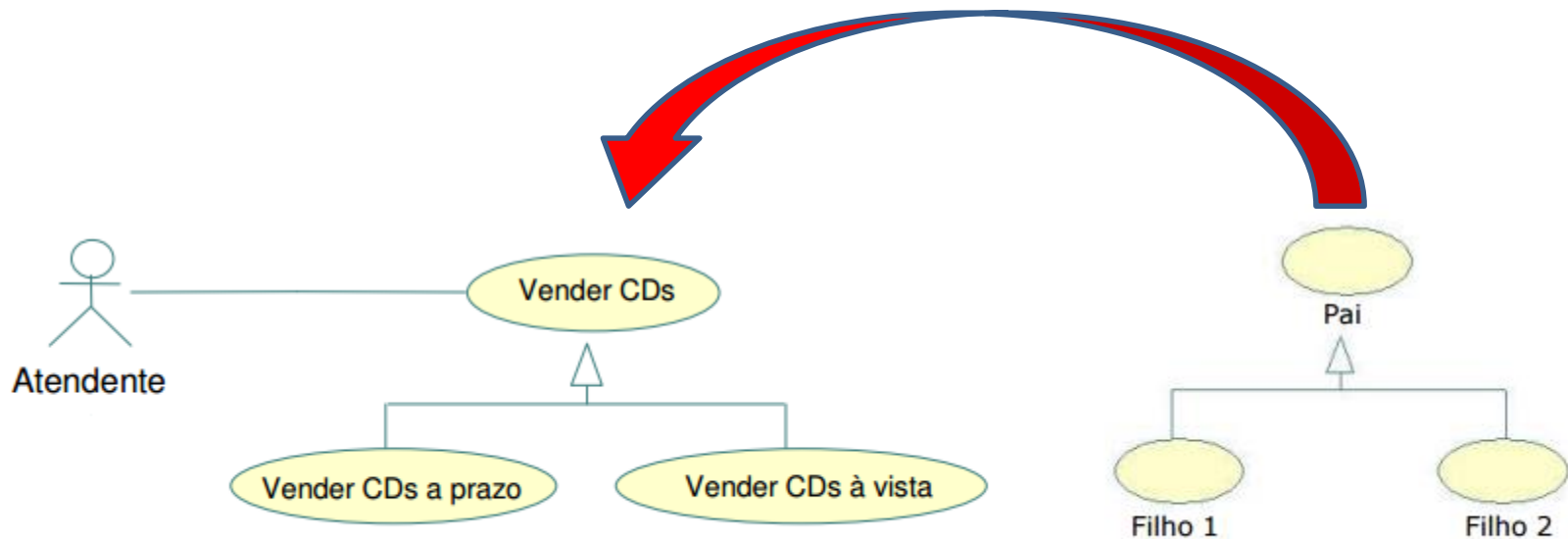
A direção do relacionamento é sempre do **generalizador** (aqui o caso de uso **B**) para o **generalizado** (caso de uso **C**).



# Modelagem de Casos de Uso

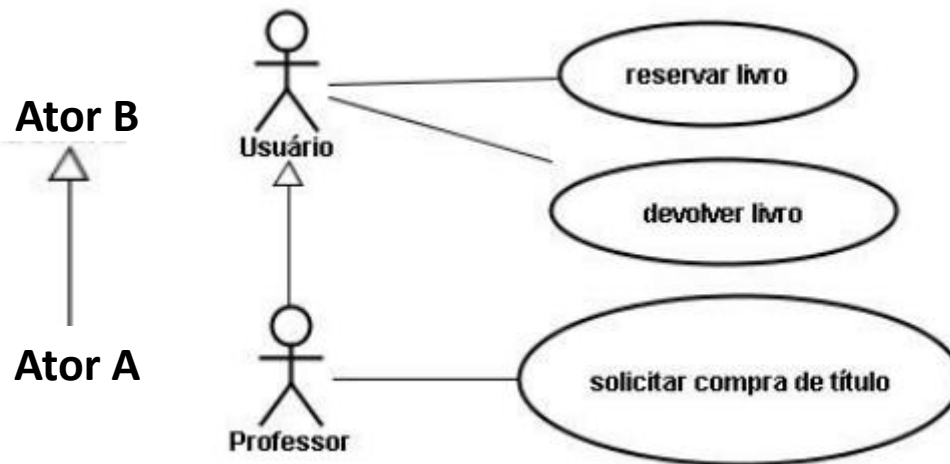
## Exemplo de Generalização/Especialização

No caso de uso **Vender CDs** representa um caso de uso genérico, que tem dois casos de usos mais específicos: **Vender CDs a prazo** e **Vender CDs à vista**.



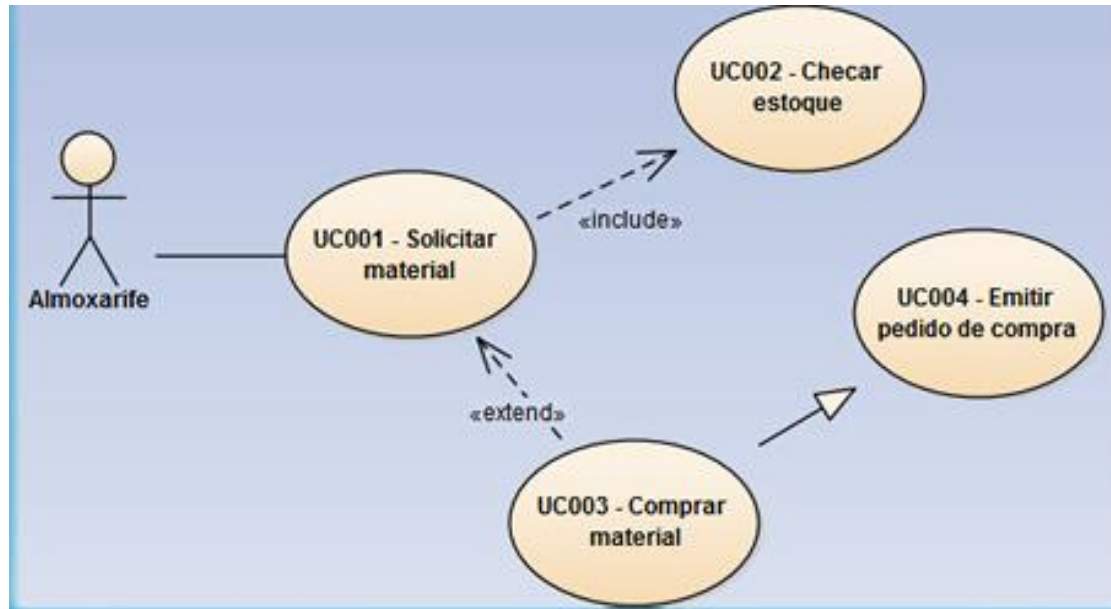
## Associações entre atores

- **Generalização/Especialização (*Generalization*):**
  - ✓ Pode ser representado, também, um tipo de relacionamento entre atores. Este relacionamento é de generalização.
  - ✓ Uma generalização de um ator **A** para um ator **B** indica que **A** pode se comunicar com os mesmos casos de uso que **B**.



# Modelagem de Casos de Uso

## Exemplo:



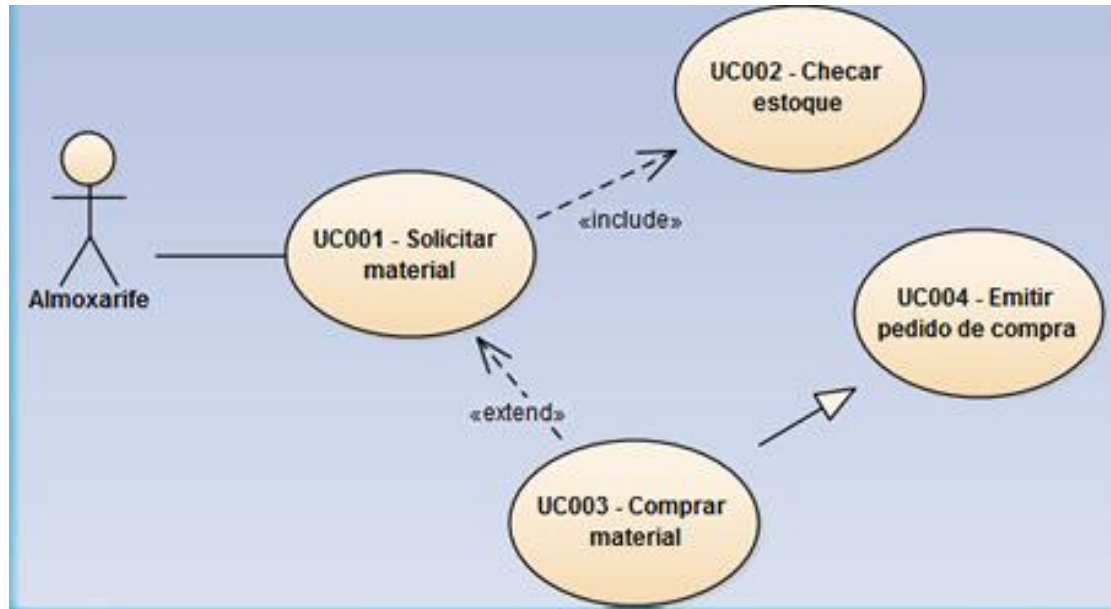
## Include

O caso de uso “Solicitar Material” faz include no caso de uso “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.

# Modelagem de Casos de Uso

## Exemplo:



## Extend

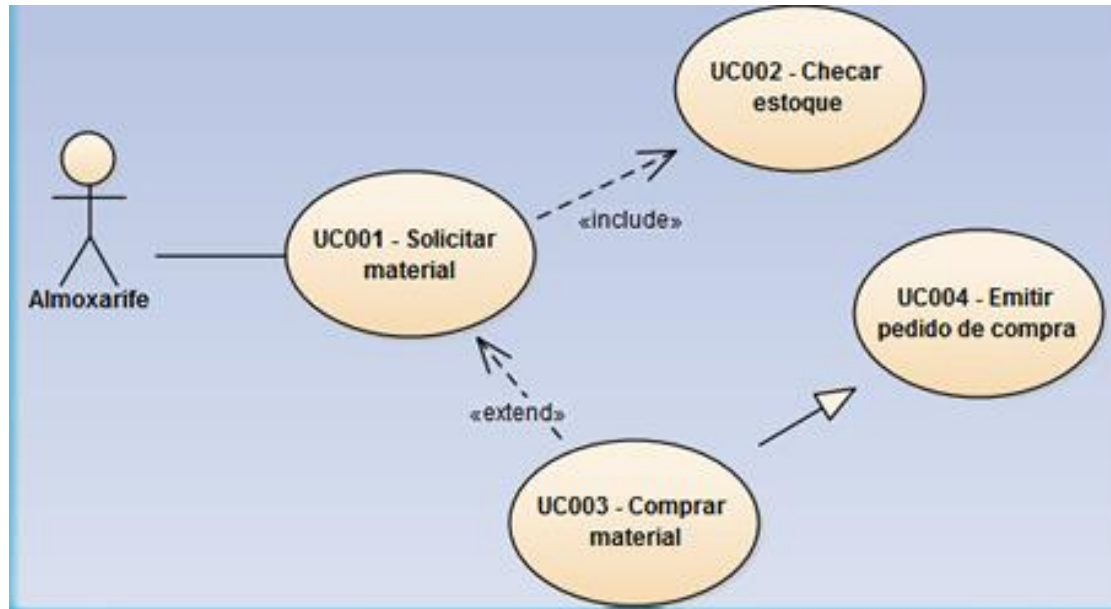
O caso de uso “Comprar Material” estende o caso de uso “Solicitar Material”. Isso se dá porque quando houver a solicitação de material, **caso o material não exista em estoque** (após consulta via o caso de uso “Checar estoque”) **poderá** ser solicitado a compra do item.

Mas também poderá não ser solicitada a compra, pois o item pode existir em estoque. Se **poderá** ser solicitada a compra (e não **sempre** será solicitada a compra) o relacionamento correto é o extend.



# Modelagem de Casos de Uso

## Exemplo:



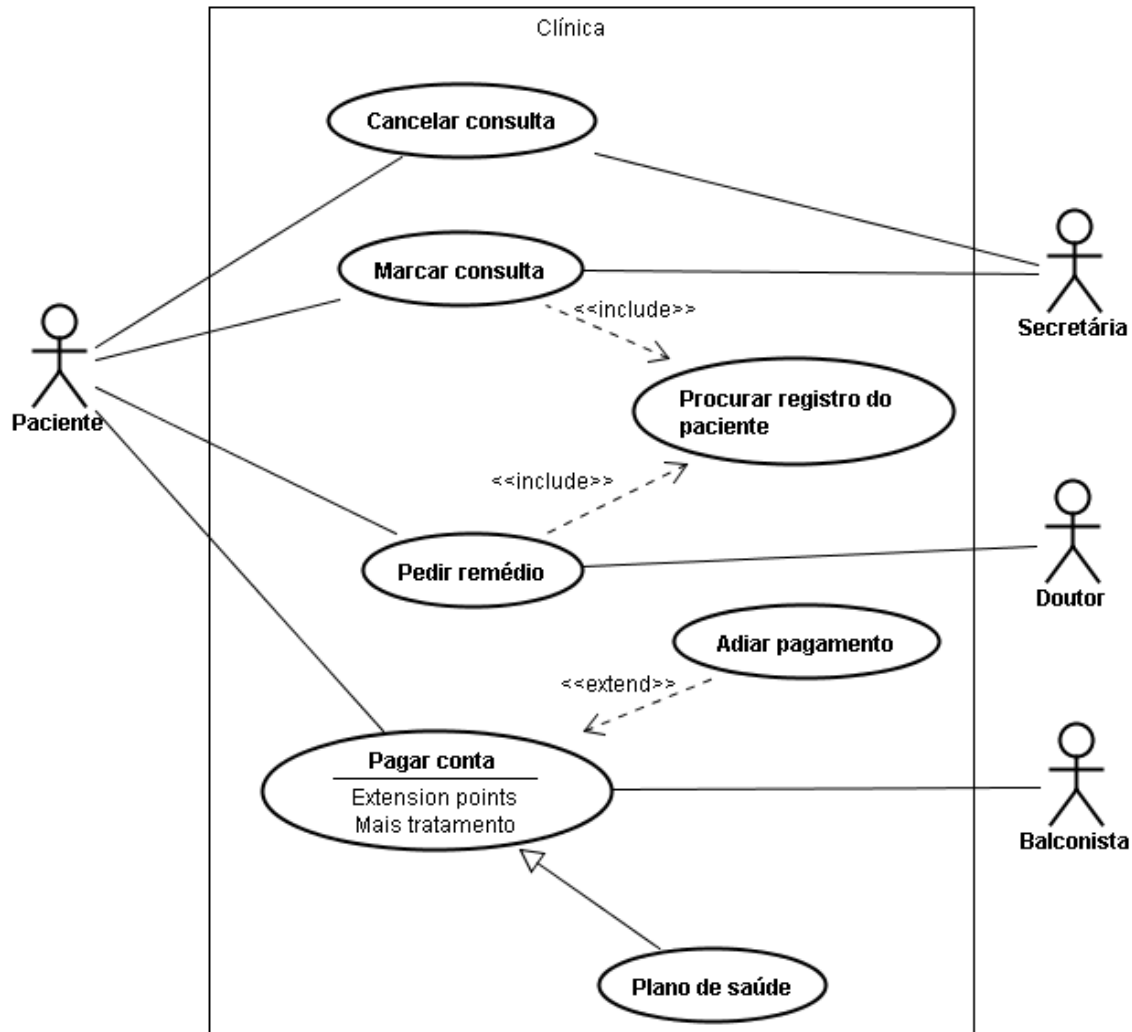
## Generalization

O caso de uso “Comprar Material” generaliza o caso de uso “Emitir pedido de compra”. Isso se dá porque no caso de uso “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.

Dessa forma, não justifica-se duplicar a especificação pertinente em outro caso de uso, basta **reaproveitar** o que já está pronto mas generalizado a ponto de poder ser aproveitado por alguém que o especialize.

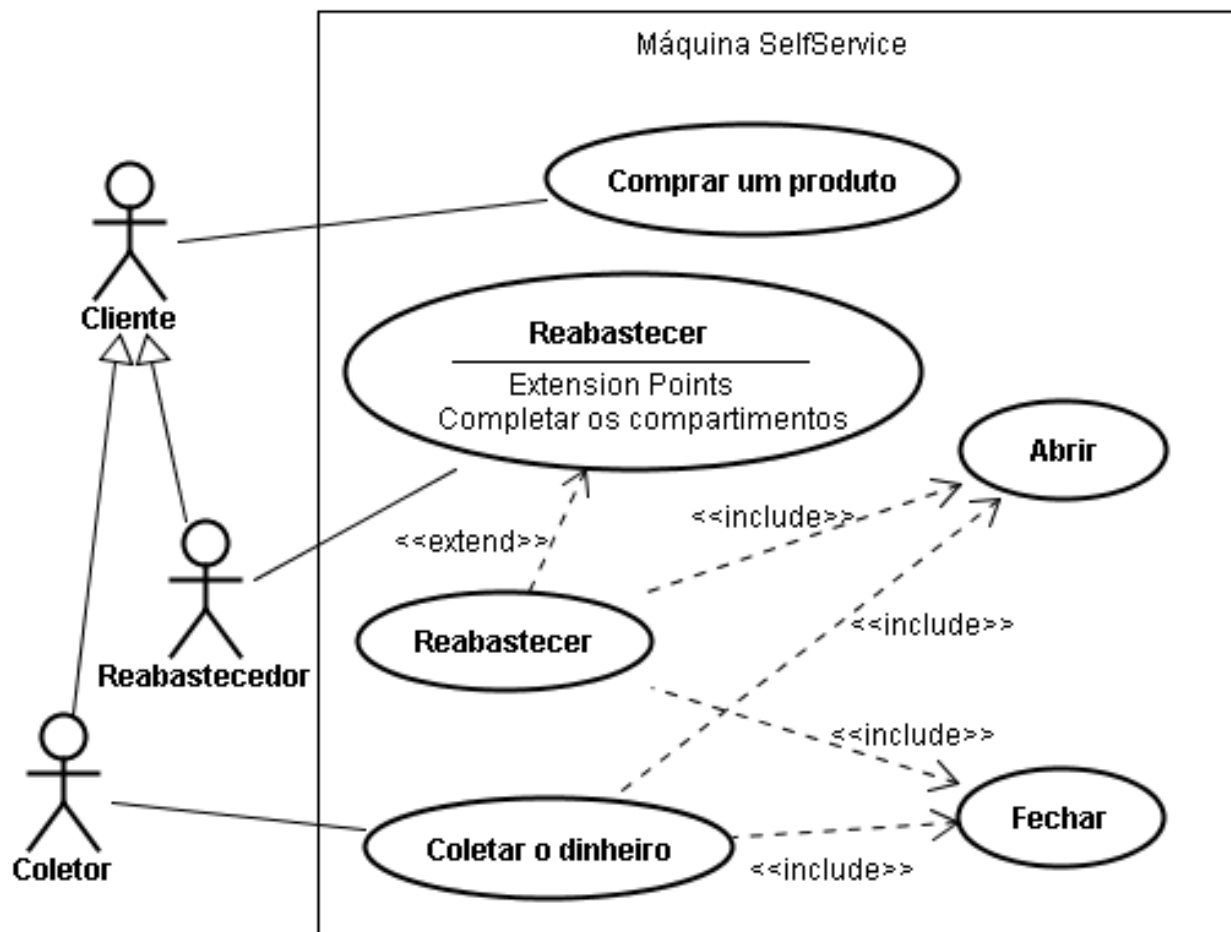
# Modelagem de Casos de Uso

## Exemplo:



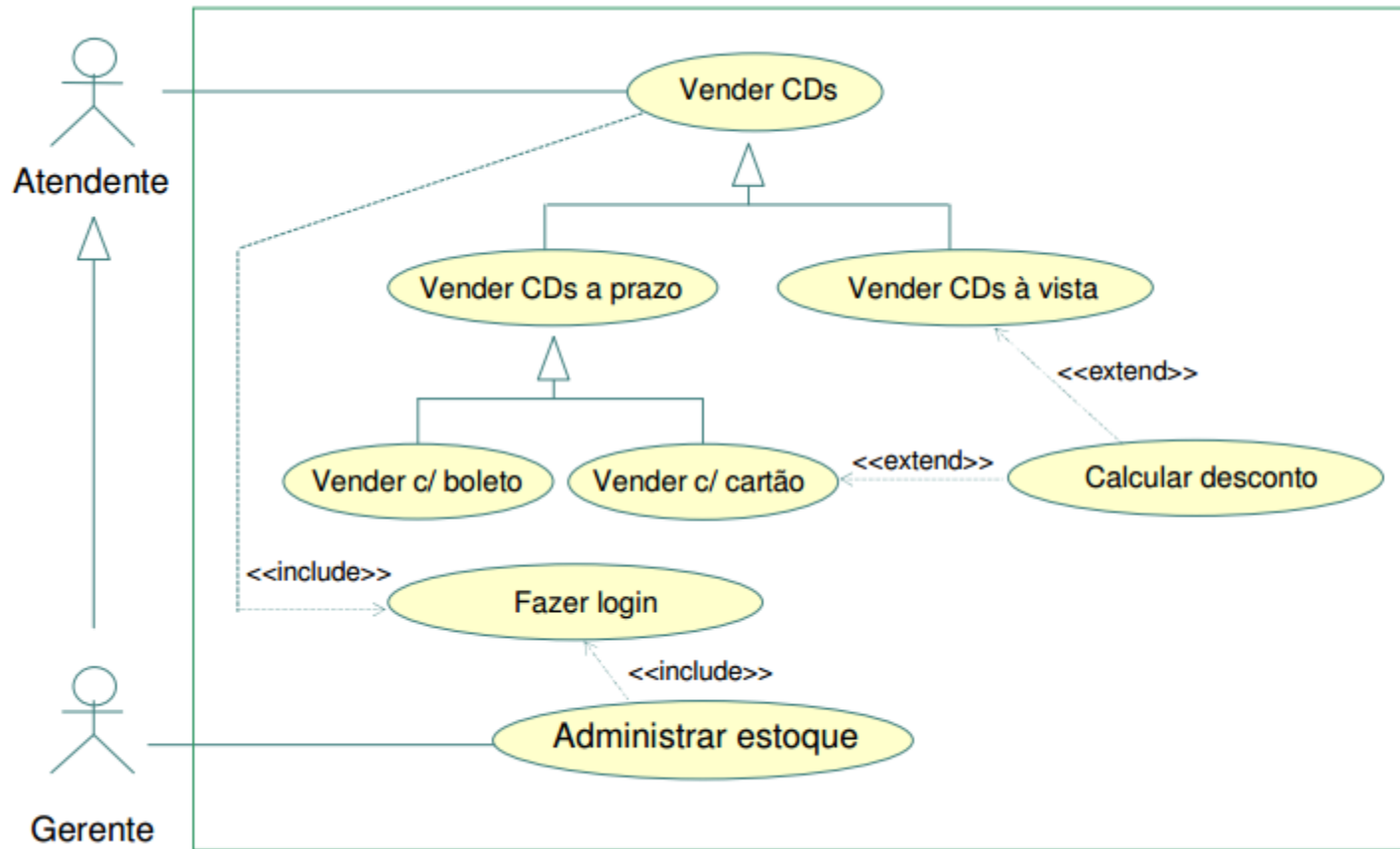
# Modelagem de Casos de Uso

## Exemplo:



# Modelagem de Casos de Uso

## Exemplo:



# Ferramentas para fazer Diagramas de Caso de Uso

---

## □ Astah

<http://www.astah.net>




# Astah

---

**Astah Community** é uma ferramenta gratuita voltada para a modelagem de diagramas UML( Unified Modeling Language).

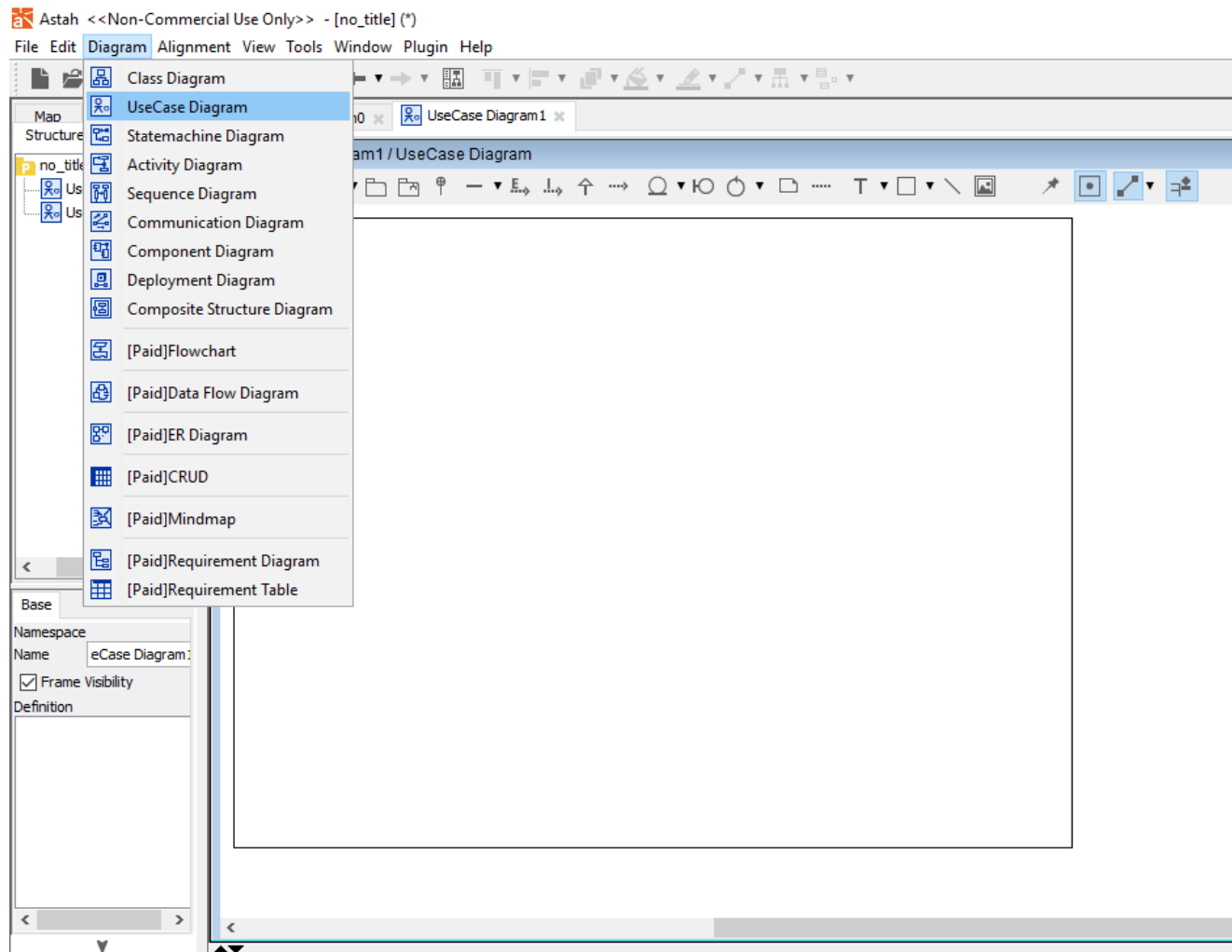
Além do Astah Community, existem outras três versões:

- **Astah UML**
  - **Astah Professional**
  - **Astah Share.**
- 
- licença comercial**

A ferramenta Astah Community é conhecida por sua praticidade e simplicidade em elaborar diagramas como por exemplo: diagramas de classes, casos de uso, de sequencia, de atividades , etc.

# Astah

Para fazer um Diagrama de caso de uso :  
Vá em no menu Diagram -> UserCase Diagram



# Astah



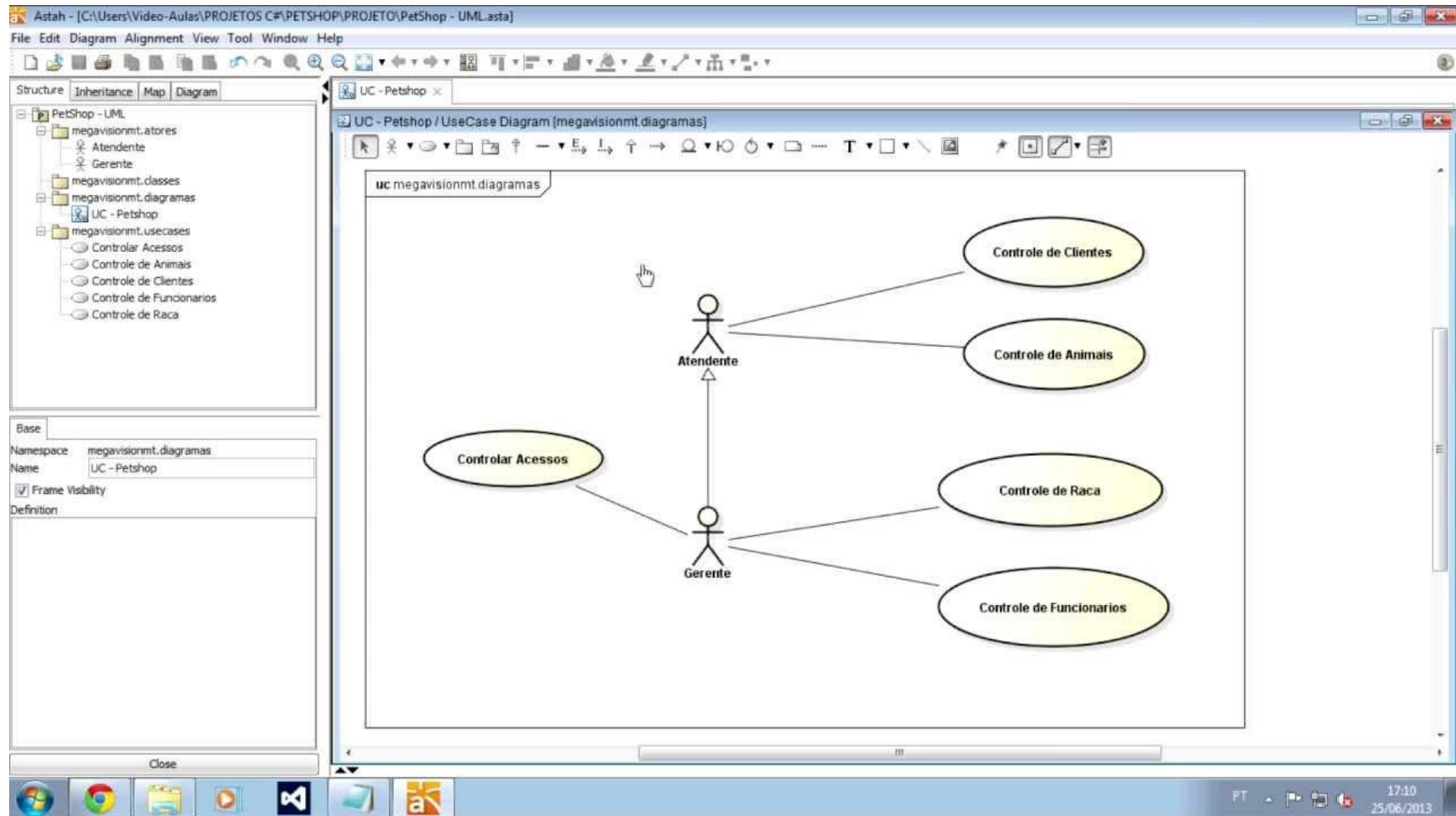
A seguir, citaremos a funcionalidade dos elementos desse menu superior:

- 1 - seleção
- 2 - cria ator
- 3 - cria caso de uso
- 4 - cria pacote
- 5 - cria ligação do tipo nest
- 6 - cria associação (clique na seta ao lado para escolher o tipo de associação)
- 7 - cria extend
- 8 - cria include
- 9 - cria generalização
- 10 - cria dependência
- 11 - cria nota
- 12 - liga a nota a um elemento do caso de uso
- 13 - texto
- 14 - cria retângulo
- 15 - cria linha
- 16 - adiciona figura
- 17 - ativa modo de bloqueio de seleção
- 18 - quando ativo a associação sempre ficará no centro ou no final do caso de uso. Desativado, a associação ficará em qualquer lugar
- 19 - muda o estilo da linha
- 20 - ativa ou desativa modo de bloqueio de profundidade



# Astah

- Exemplo de UC Diagram utilizando o sistema Astah

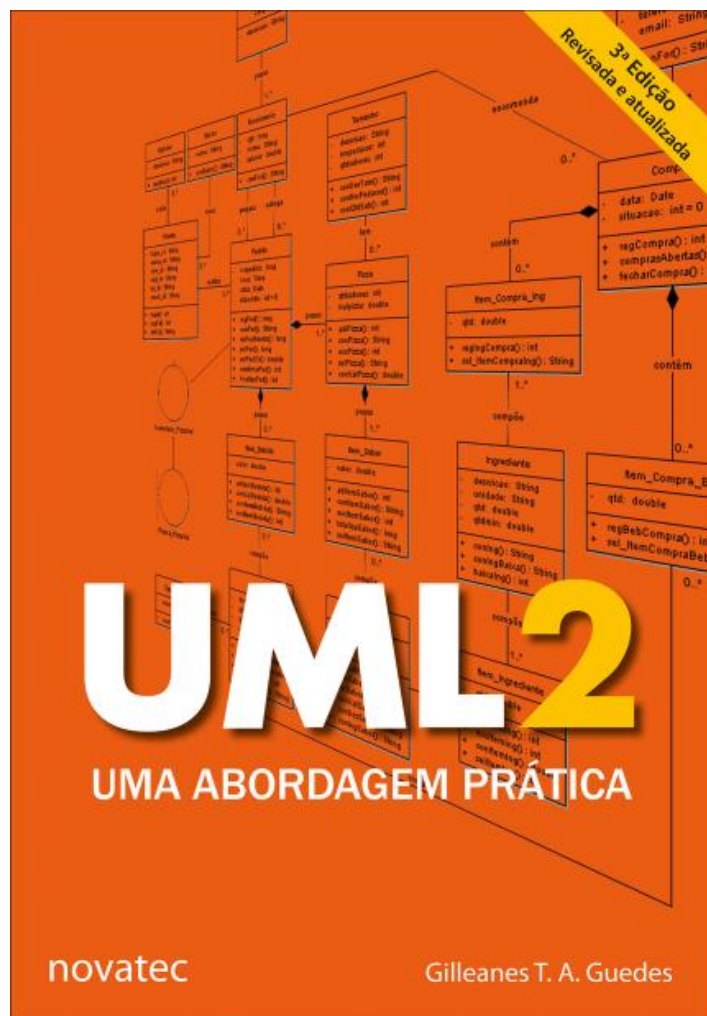


# Astah

- Exemplo de Descrição de Caso de Uso utilizando o sistema Astah

incluir aluno / UseCase Description [use case]	
ITEM	VALUE
UseCase	incluir aluno
Summary	descreve a funcionalidade de incluir um aluno
Actor	secretaria
Precondition	aluno não pode constar na base de dados (não pode incluir algo que já existe)
Postcondition	aluno incluído na base de dados
Base Sequence	1) secretaria inclui os seguintes dados do aluno: (nome, prontuario*) 2) sistema retorna sucesso na operação
Branch Sequence	
Exception Sequence	1) se aluno existir na BD emitir mensagem com o erro. 2) se campo obrigatório não for preenchido, emitir mensagem com o erro.
Sub UseCase	
Note	

- Bibliografia**



## Referencias

- Boock, G. and Rumbaugh, J. **The Unified Modeling Language User Guide** . Addison-Wesley, 1999
- Arlow, J. and Neustadt, I. **UML 2 and the Unified Process: Practical Object-Oriented Analysis and Design**, 2nd Edition, The AddisonWesley Object Technology Series, 2005.
- Rumbaugh, J.; Jacobson, I. and Booch , G. **The Unified Modeling Language Reference Manual, 2nd Edition**, The Addison-Wesley Object Technology Series, 2004.
- Boock, G.; Rumbaugh, J. and Jacobson, I; **Unified Modeling Language User Guide, 2nd Edition**, The Addison-Wesley Object Technology Series, 2005.
- Jacobson, I; Boock, G. and Rumbaugh, J., **Unified Software Development Process**, Addison-Wesley, Janeiro 1999.
- Larman, C. Applying UML and Patterns: **An Introduction to Object Oriented Analysis and Design Prentice-Hall**, New Jersey - USA, 1997
- Bezerra, E. **Princípios de Análise e Projeto com a UML**, ed. Campus Elsevier. 2003.