

# Engenharia de Software

**Aula:**  
**A Linguagem UML.**  
**Visões de um Sistema.**  
**Diagramas UML.**



Prof. Anderson Augusto Bosing

# A Linguagem UML

A Unified Modeling Language (UML, Linguagem de Modelagem Unificada) é “uma linguagem-padrão para descrever/documentar projeto de software.

A UML pode ser usada para visualizar, especificar, construir e documentar os artefatos de um sistema de software.



# A Linguagem UML

## Motivação

Em outras palavras, assim como os arquitetos criam plantas e projetos para serem usados por uma empresa de construção, os arquitetos de software criam diagramas UML para ajudar os desenvolvedores de software a construir o software. Se você entender o vocabulário da UML (os elementos visuais do diagrama e seus significados), poderá facilmente entender e especificar um sistema e explicar o projeto desse sistema para outros interessados.



# A Linguagem UML

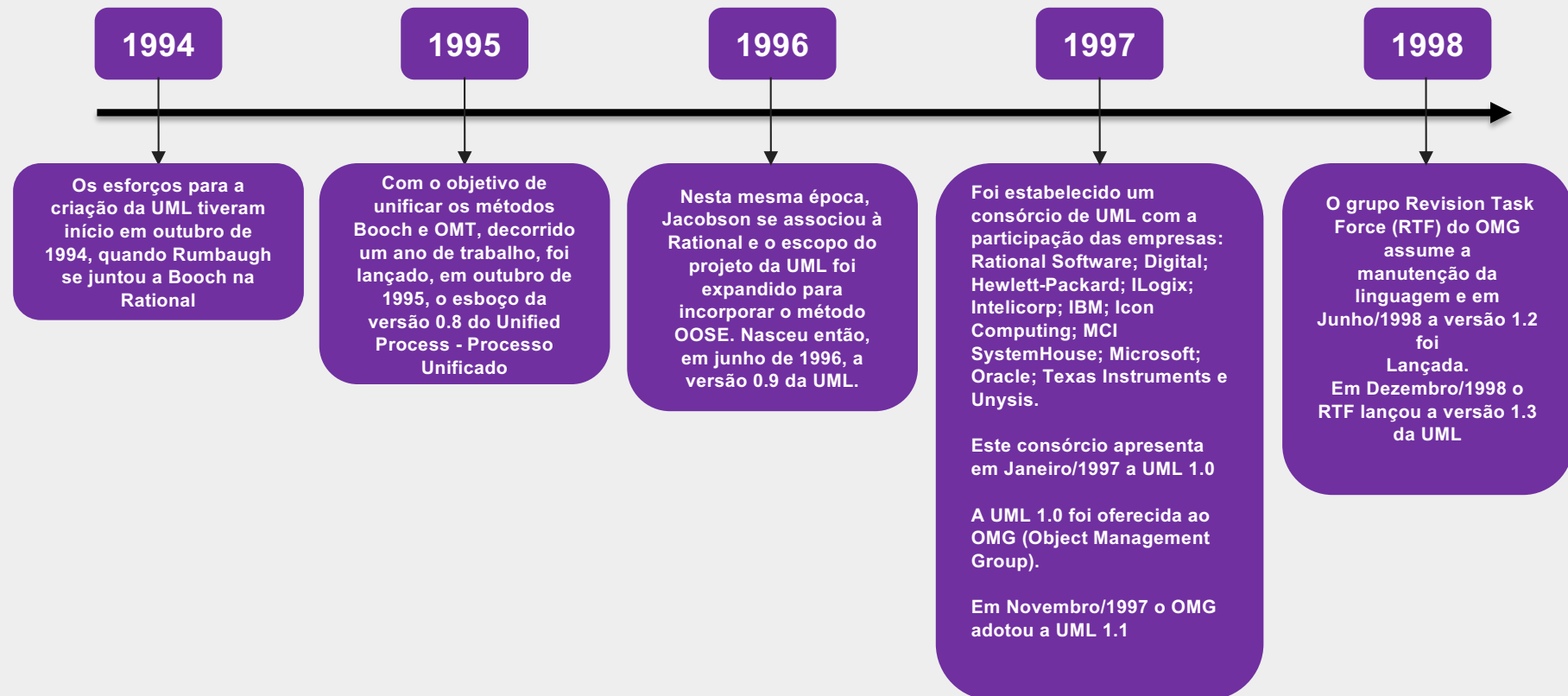
A construção da UML teve muitos contribuintes, mas os principais atores no processo foram Grady Booch, James Rumbaugh e Ivar Jacobson.

Em 1997 a UML foi aprovada como padrão pelo OMG(Object Management Group).

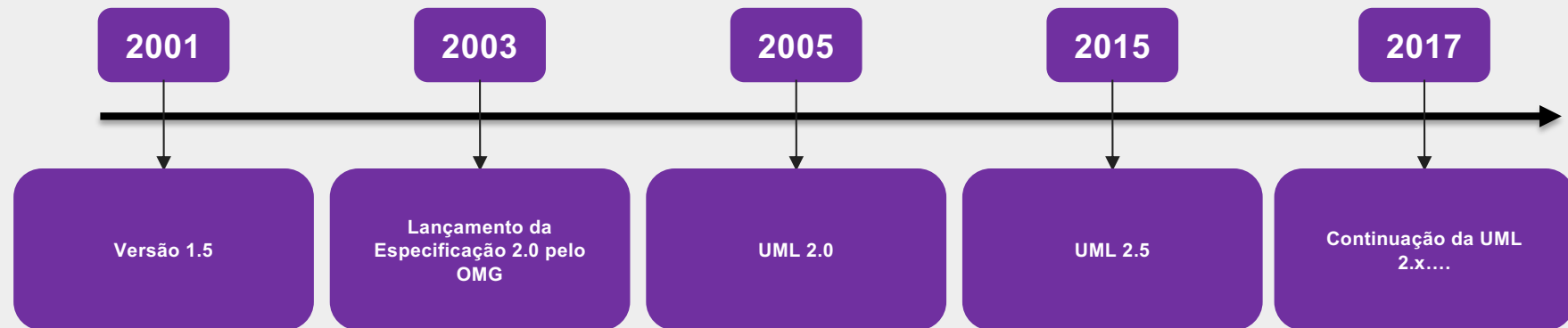
Desde então, a UML tem tido grande aceitação pela comunidade de desenvolvedores de sistemas.



# A Linguagem UML Timeline



# A Linguagem UML Timeline

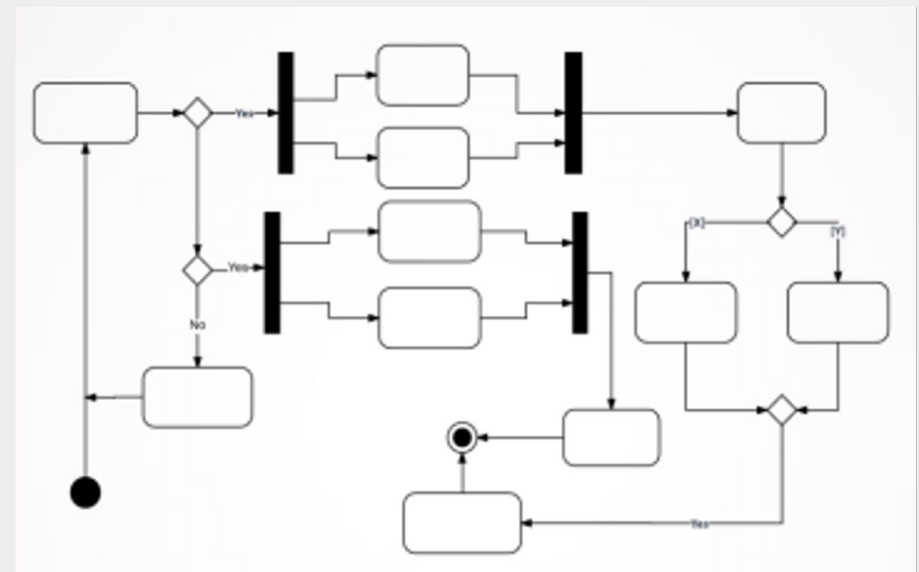


# Visão Geral da UML



A UML é uma linguagem visual para modelar sistemas orientados a objetos.

Por meio dos elementos gráficos definidos nesta linguagem pode-se construir diagramas que representam diversas perspectivas de um sistema.



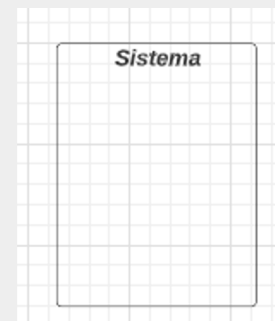
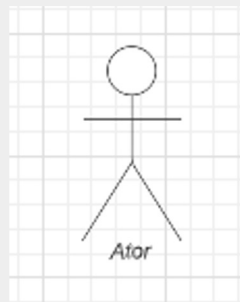
# Visão Geral da UML



Cada **elemento gráfico** da UML possui uma sintaxe e uma semântica.

A **sintaxe** de um elemento corresponde à forma predeterminada de desenhar o elemento.

A **semântica** define o que significa o elemento e com que objetivo ele deve ser utilizado.





# Visão Geral da UML

A UML é independente tanto de linguagens de programação quanto de processos de desenvolvimento.

Esse é um fator importante para a utilização da UML, pois diferentes sistemas de software requerem abordagens diversas de desenvolvimento.



# Visão Geral da UML

O desenvolvimento de um software complexo demanda que seus desenvolvedores tenham a possibilidade de examinar e estudar esse sistema a partir de perspectivas diversas.

Os autores da UML sugerem que um sistema pode ser descrito por cinco visões:

- ☐ Visão de Casos de Uso.
- ☐ Visão de Projeto.
- ☐ Visão de implementação.
- ☐ Visão de Implantação.
- ☐ Visão de Processo.



# Visão Geral da UML

## Visão de Caso de Uso

Descreve o sistema de um ponto de vista externo como um conjunto de interações entre o sistema e os agentes externos ao sistema.

Esta visão é criada em um estágio inicial e direciona o desenvolvimento das outras visões do sistema.



# Visão Geral da UML

## Visão de Projeto(Lógica)

**Enfatiza as características do sistema que dão suporte, tanto estrutural quanto comportamental, às funcionalidades externamente visíveis do sistema.**

**Ligada ao problema do negócio.**



# Visão Geral da UML

## Visão de Implementação

**Abrange o gerenciamento de versões do sistema, construídas pelo agrupamento de módulos (componentes) e subsistemas.**



# Visão Geral da UML

## Visão de Implantação

**Corresponde à distribuição física do sistema em seus subsistemas e à conexão entre essas partes.**



# Visão Geral da UML

## Visão de Processo

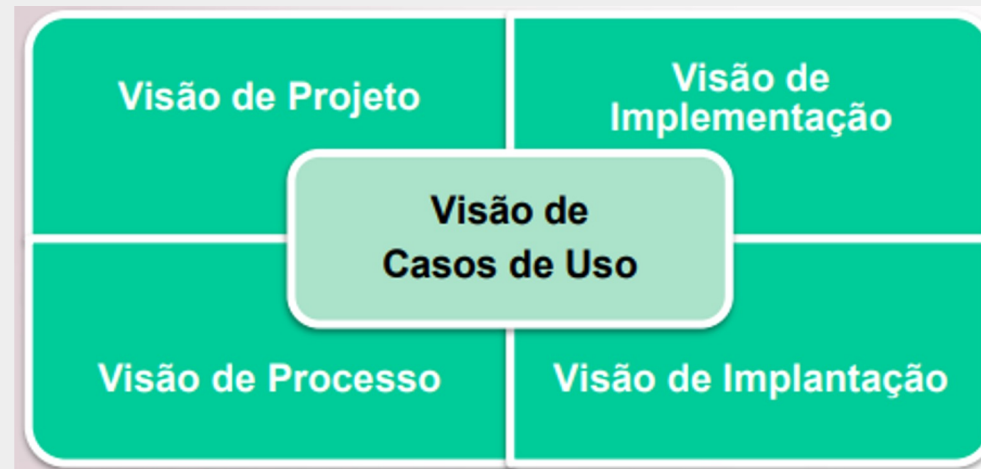
Esta visão enfatiza as características de concorrência (paralelismo), sincronização e desempenho do sistema.



# Visão Geral da UML

## Visão da UML

Dependendo das características do sistema, nem todas as visões precisam ser construídas.





# Visão Geral da UML

## Visão da UML

Um processo de desenvolvimento que utilize a UML envolve a criação de diversos documentos.

Esses documentos podem ser textuais ou gráficos.

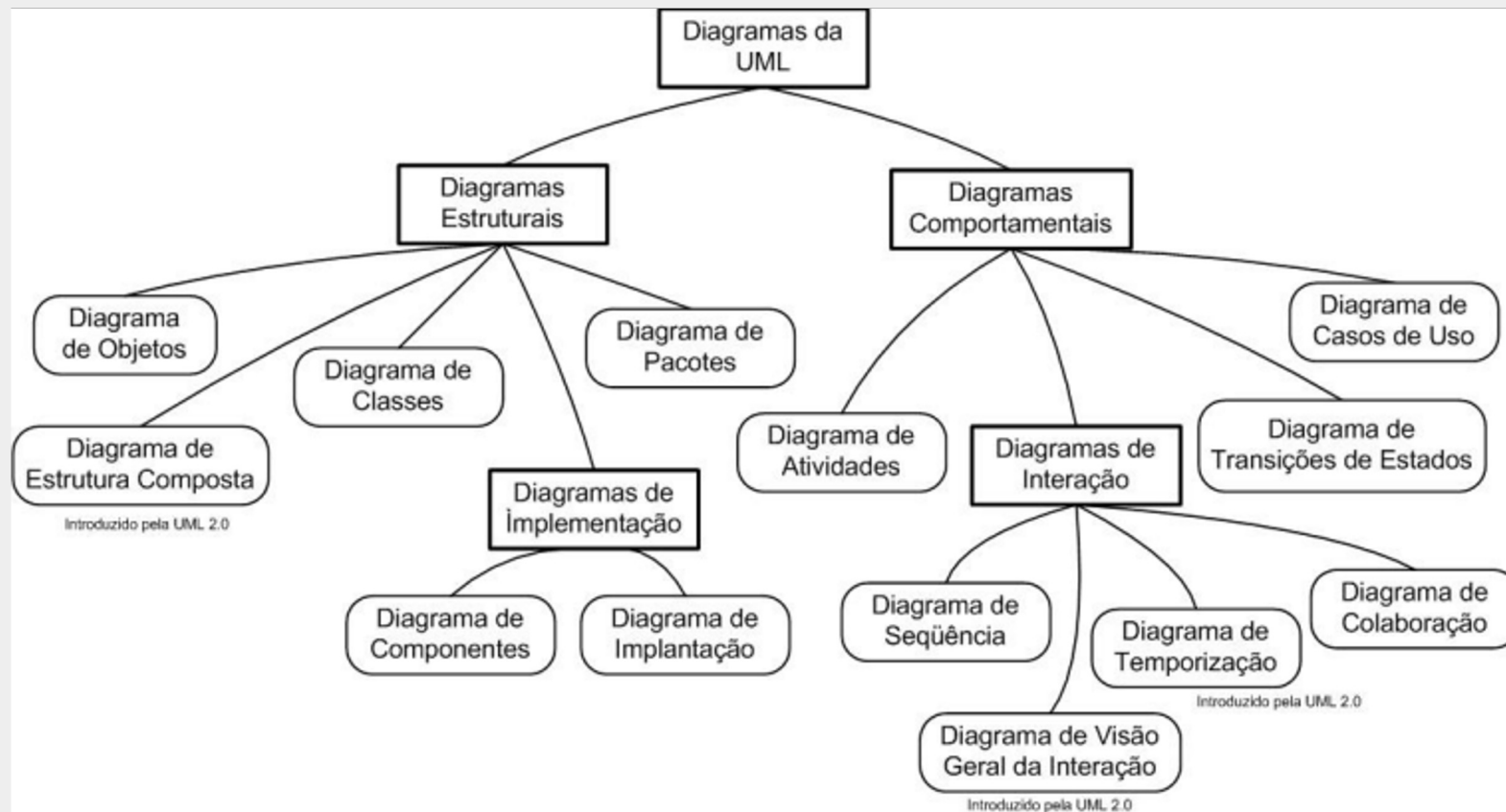
Eles são denominados artefatos de software, ou simplesmente artefatos.

São os artefatos que compõem as visões do sistema.



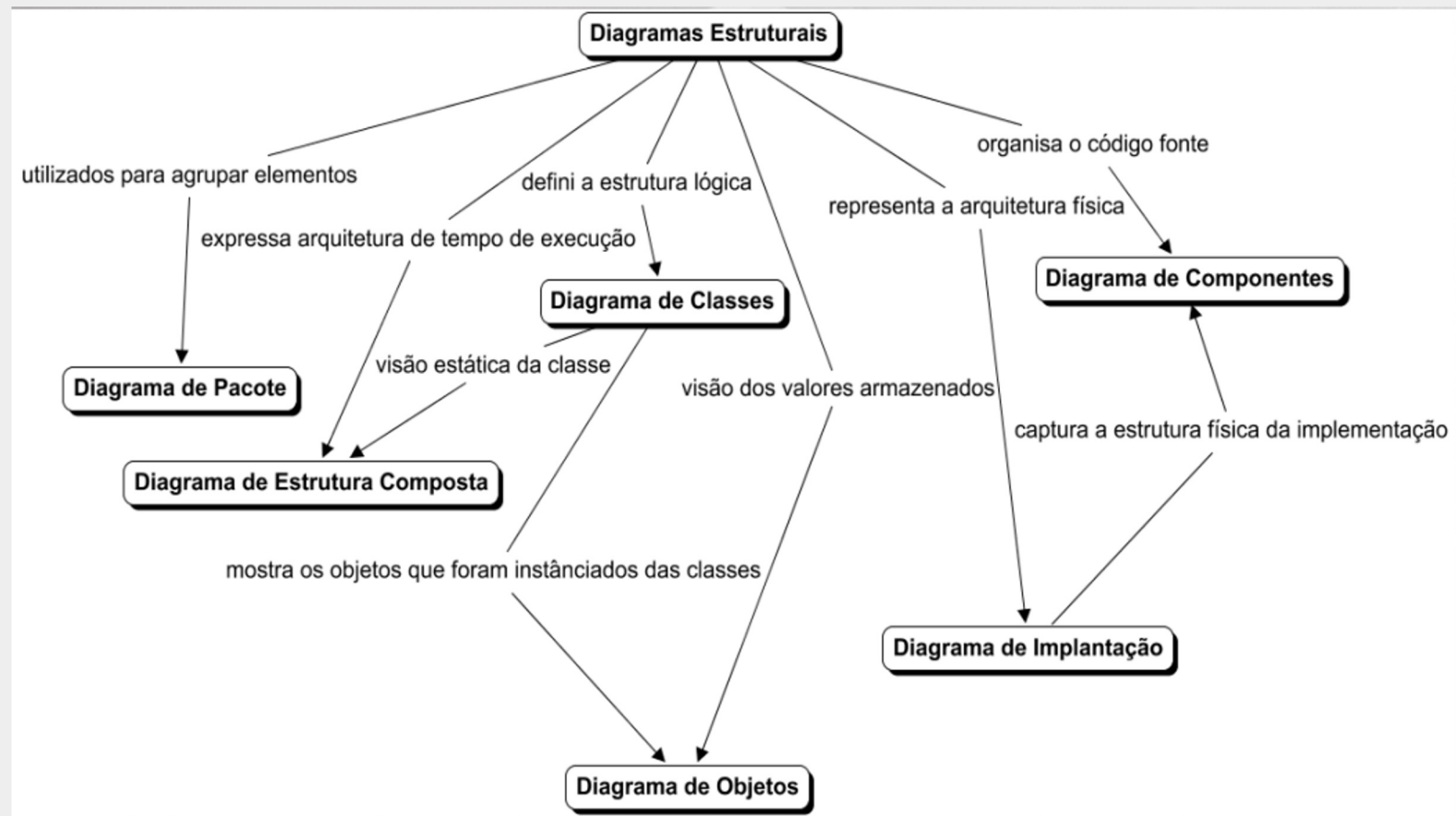
# Visão Geral da UML

## Artefatos da UML



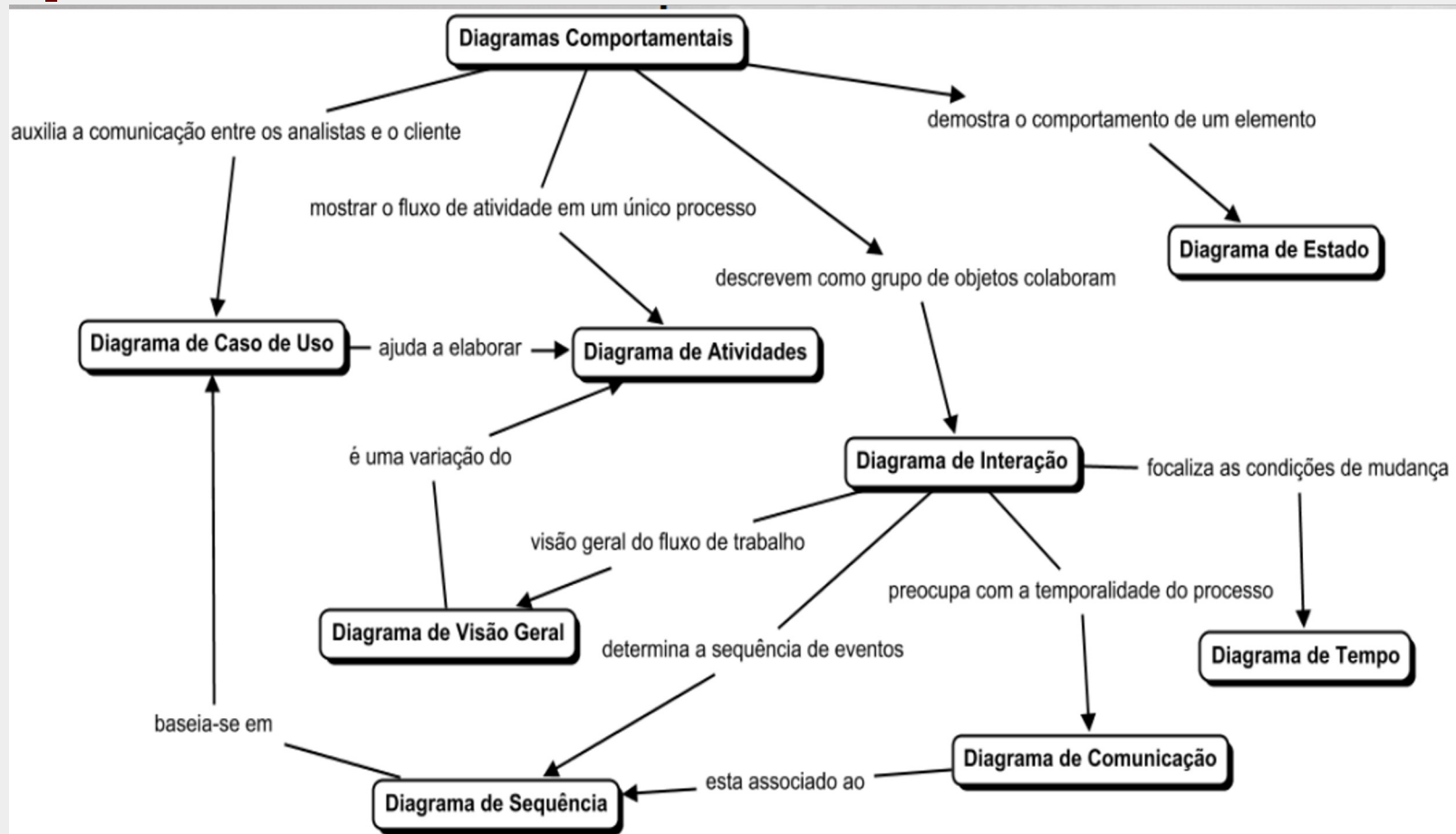
# Visão Geral da UML

## Mapa Conceitual - Artefatos da UML



# Visão Geral da UML

## Mapa Conceitual - Artefatos da UML



# A Linguagem UML

## Site, Certificações e o futuro

<https://www.uml.org/>



# Análise e Projetos de Sistemas

**Aula:**  
**Modelo de Caso de Uso**



Prof. Anderson Augusto Bosing

**Sistemas não existem de forma isolada. Existem interações com humanos ou máquinas.**

# Modelagem de Caso de Uso - Histórico

- Foi originalmente desenvolvida por Jacobson et al. (1993).
- Foi incorporada ao primeiro release da UML (RUMBAUGH et al., 1999).



# **Modelo de Casos de Uso(MCU)**

**O MCU representa os possíveis usos de um sistema da maneira como são percebidos por um observador externo a este sistema.**

**Cada um desses usos está associado a um ou mais requisitos funcionais identificados para o sistema.**

**É um modelo de análise que representa um refinamento dos requisitos funcionais.**

**Possui diversos componentes: casos de uso, atores e relacionamentos entre eles.**

# O que é Modelagem de Caso de Uso?

- Faz a ligação entre as necessidades dos envolvidos para os requisitos de software.
- Define limites claro para um sistema.
- Captura e comunica o comportamento desejado do sistema.
- Identifica quem ou o que interage com o sistema.
- Valida/Verifica os requisitos.
- Um instrumento de planejamento.

# O que é um Caso de Uso ?

**Um caso de uso define uma sequência de ações a serem executadas pelo sistema para produzir um resultado de valor observável para um ator.(RUP)**

**Usamos casos de uso para captar os comportamentos pretendidos de um sistema, sem especificar como esse comportamento é implementado.**

# **Casos de Uso são comumente identificados por nomes ou identificadores.**

**Todo caso de uso possui um nome que o identifica e diferencia dos demais casos de uso do sistema.**

**O nome é uma sequência de caracteres de texto e deve ser único no pacote que o contém.**

**No geral, os nomes são expressões verbais ativas, que nomeiam um comportamento específico do sistema.**

**Exemplos: Registrar Venda, Fazer Pedido, Manter Usuários, Manter Produtos.**

# **Casos de Uso são comumente identificados por nomes ou identificadores.**

**Os identificadores representam através de um código o caso de uso a que se referenciam.**

**Exemplo:**

**UC.001 - Registrar Venda**

**UC.002 - Fazer Pedido**

**UC.003 - Manter Usuários**

**UC.004 - Manter Produtos**

# O que é um Caso de Uso ?

Um caso de uso é a especificação de uma sequência completa de interações entre um sistema e um ou mais agentes externos a esse sistema.

Representa um relato de uso de certa funcionalidade do sistema em questão, sem revelar a estrutura e o comportamento internos desse sistema.

# O que é um Caso de Uso ?

Cada caso de uso de um sistema se define pela descrição narrativa das interações que ocorrem entre o elemento externo e o sistema.

**A UML não define uma estrutura textual a ser utilizada na descrição de um caso de uso.**

Há vários estilos de descrição propostos para definir casos de uso.

# Casos de Uso contém Requisitos de Software

Cada caso de uso:

- Descreve ações que o sistema faz para entregar algo de valor para um agente.
- Mostra a funcionalidade do sistema que o agente usa.
- Modela um diálogo entre o sistema e os agentes.
- É um completo e significativo fluxo de eventos da perspectiva de um agente em particular.



# Benefícios dos Casos de Uso

- **Dá um contexto para os requisitos**

**Coloca os requisitos do sistema em sequências lógicas.**

**Ilustra o porque o sistema é necessário.**

**Ajuda a verificar se todos os requisitos foram capturados.**

- **São fáceis de entender.**

**Usam terminologia que clientes e usuários entendem.**

**Descreve a história concreta de uso do sistema.**

**Verifica o entendimento dos envolvidos.**

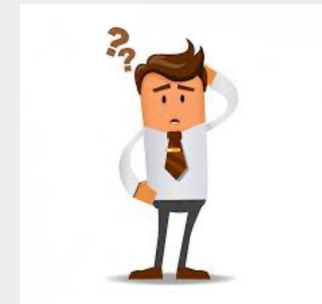
- **Facilita o acordo com os clientes.**

- **Facilita o reuso: teste, documentação e design.**

# Elementos do Diagrama de Caso de Uso

**“Um caso de uso é a especificação de uma sequência completa de interações entre um sistema e um ou mais **agentes externos** a esse sistema.”**

**O que são os agentes externos? Onde vivem? Do que se alimentam ?**



# Atores

Qualquer elemento externo ao sistema que interage com o mesmo é denominado ator.

**“Externo”** nessa definição indica que atores não fazem parte do sistema.

**“Interage”** significa que um ator troca informações com o sistema (envia ou recebe informações).

# Atores

Podem ser Categorizados em:

Um **ator humano** é uma pessoa física, que no diagrama deve possuir como nome o papel que a pessoa executa no contexto empresarial onde o sistema será utilizado. Por exemplo: Cliente, Fornecedor, Atendente.

Um **ator sistêmico** é um sistema, ou módulo de um sistema, ou componente de um sistema, que realizará a execução da funcionalidade que está especificada pelo caso de uso. No diagrama deve possuir seu nome de fato (se o ator é o sistema “Disparador de rotinas batch” por exemplo, este deve ser o nome do ator sistêmico).

# Atores

## Exemplos de Atores:

- ❑ **Cargos (Ex. Empregado, Cliente, Gerente, Vendedor).**
- ❑ **Organizações ou divisões de uma organização (Ex. Fornecedor, Administradora de Cartões, Almoxarifado ).**
- ❑ **Outros sistemas de software (Ex. Sistema de Cobrança, Sistema de Estoque de Produtos).**
- ❑ **Equipamentos com os quais o sistema deve se comunicar (Ex. Leitora de Código de Barras, Sensor).**

# Atores

**Um ator corresponde a um papel representado em relação ao sistema.**

## **Exemplos:**

**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 um banco que realiza a manutenção de um caixa eletrônico, mas também pode ser o Cliente do banco que realiza o saque.**

# Atores

**É uma boa prática de modelagem fazer com que o nome dado a esse ator lembre o seu papel, em vez de lembrar quem o representa.**

**Exemplos de bons nomes para atores:**

**✓ Cliente, Estudante, Fornecedor etc.**

**Exemplos de maus nomes para atores:**

**✓ João, Fornecedora, ANVISA etc.**



# Atores

Um ator pode participar de muitos casos de uso (situação muito comum na prática).

Do mesmo modo, um caso de uso pode envolver a participação de vários atores, nesses casos, os atores podem ter duas classificações:

**Primários** ou **Secundários**.

# Atores

**Um ator primário é aquele que inicia uma sequência de interações de um caso de uso.**

**São eles os agentes externos para os quais o caso de uso traz benefício direto.**

**As funcionalidades principais do sistema são definidas tendo em mente os objetivos dos atores primários.**

# Atores

**Já um ator secundário supervisiona, opera, mantém ou auxilia na utilização do sistema pelo atores primários.**

**Atores secundários existem apenas para que os atores primários possam utilizar o sistema.**

# Atores

**Exemplo:**

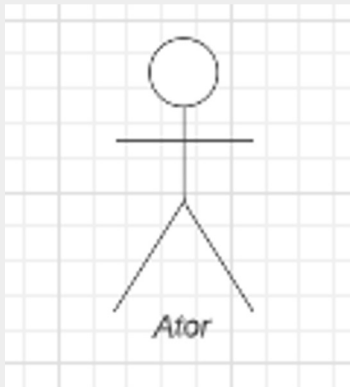
**Em um navegador de internet.**

**Para que o Usuário (ator primário) requisiite uma página ao programa, outro ator (secundário) está envolvido: o Servidor Web.**

**O ator primário Usuário é auxiliado pelo secundário, Servidor Web, uma vez que é através deste último que o primeiro consegue alcançar seu objetivo.**

# Atores

Notação de um Ator:



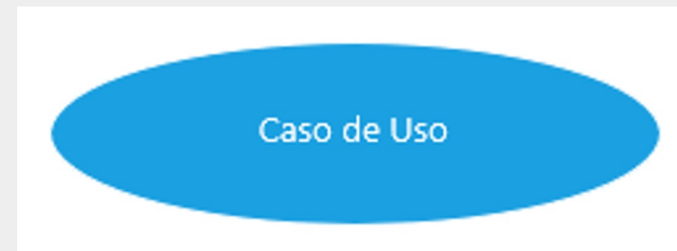
# Caso de Uso

**Um caso de uso define uma sequência de ações a serem executadas pelo sistema para produzir um resultado de valor observável para um ator.(RUP)**

**Usamos casos de uso para captar os comportamentos pretendidos de um sistema, sem especificar como esse comportamento é implementado.**

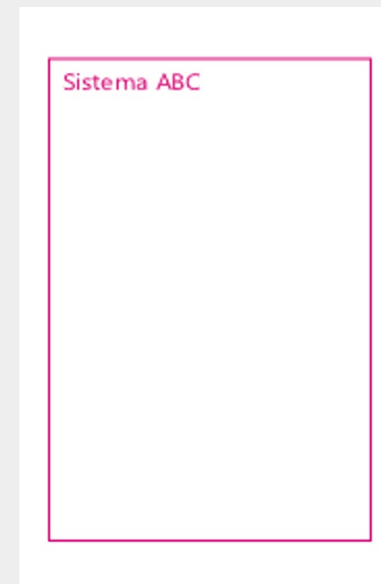
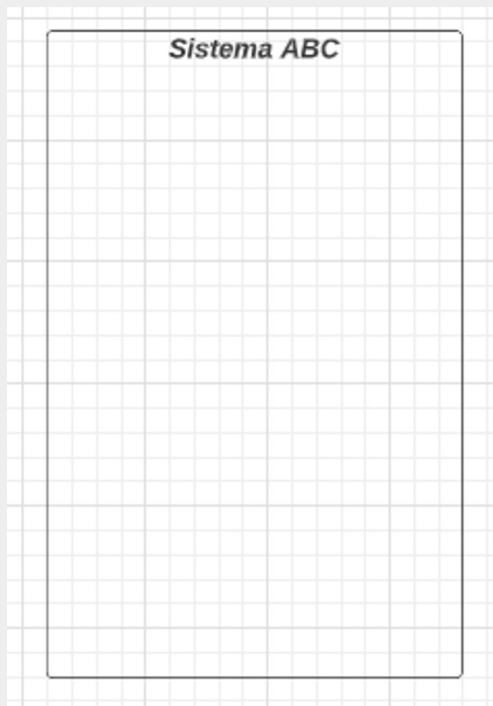
# Caso de Uso

Notação de um Caso de Uso:



# Sistema

Representa a fronteira do sistema ou componentes de sistemas e subsistemas.





# Como os Atores e Casos de Uso se relacionam ?

Um ator deve estar relacionado a um ou mais casos de uso do sistema.

Além disso, pode haver relacionamentos entre os casos de uso ou entre os atores de um sistema.

A UML define os seguintes relacionamentos para o modelo de casos de uso: comunicação, inclusão, extensão e generalização.

# Associação de Comunicação

**Um relacionamento de comunicação informa a que caso e uso o ator está associado.**

**Isso significa que esse ator interage (troca informações) com o sistema com ajuda daquele caso de uso.**

**O relacionamento de comunicação é, de longe, o mais comumente utilizado de todos.**

# Associações

Notação de um Caso de Uso:



Associação  
bidirecional



Associação  
unidirecional

# Relacionamento de inclusão.

Na modelagem UML, um relacionamento de inclusão é aquele no qual um caso de uso (o caso de uso base) inclui a funcionalidade de outro caso de uso (o caso de uso de inclusão). O relacionamento de inclusão suporta a reutilização da funcionalidade em um modelo de caso de uso.

O relacionamento de inclusão existe somente entre casos de uso. Quando dois ou mais casos de uso incluem uma sequência comum de interações, essa sequência comum pode ser descrita em outro caso de uso.

A partir daí, vários casos de uso do sistema podem incluir o comportamento desse caso de uso comum.

# Relacionamento de inclusão.

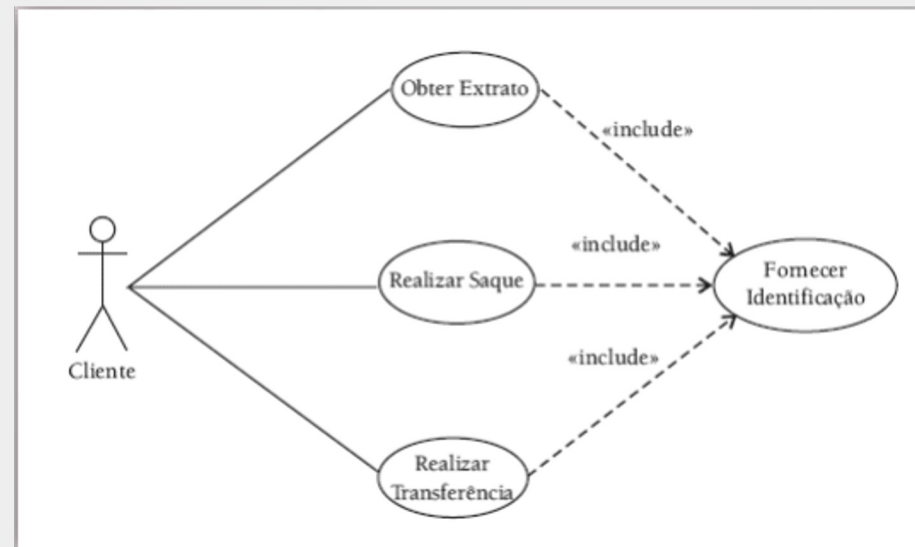
**Exemplo: Sistema bancário**

**Esses casos de uso têm uma sequência de interações em comum: a que valida a senha do cliente do banco.**

**Essa sequência de interações em comum pode ser descrita em um caso de uso Fornecer Identificação.**

**Alguns casos de uso desse sistema são Obter Extrato, Realizar Saque e Realizar Transferência.**

# Relacionamento de inclusão.



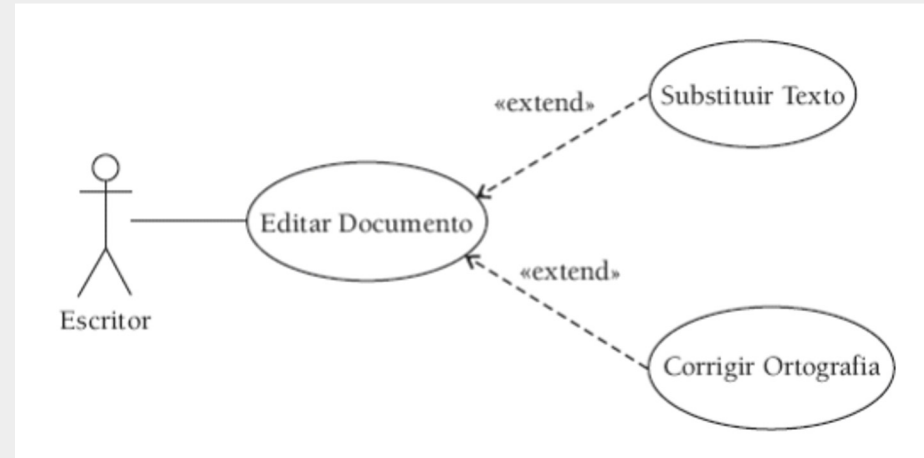
# Relacionamento de extensão.

Na modelagem UML, é possível utilizar um relacionamento de extensão para especificar que um caso de uso (extensão) estende o comportamento de outro caso de uso (base). Esse tipo de relacionamento revela detalhes sobre um sistema ou aplicativo que normalmente estão ocultos em um caso de uso.

Cada uma dessas diferentes sequências representa um comportamento eventual, ou seja, um comportamento que só ocorre sob certas condições, ou cuja realização depende da escolha do ator.

# Relacionamento de extensão.

**Mostra que os casos de uso Corrigir Ortografia e Substituir Texto têm sequências de interações que são eventualmente utilizadas quando o ator Escritor estiver usando o caso de uso Editar Documento**



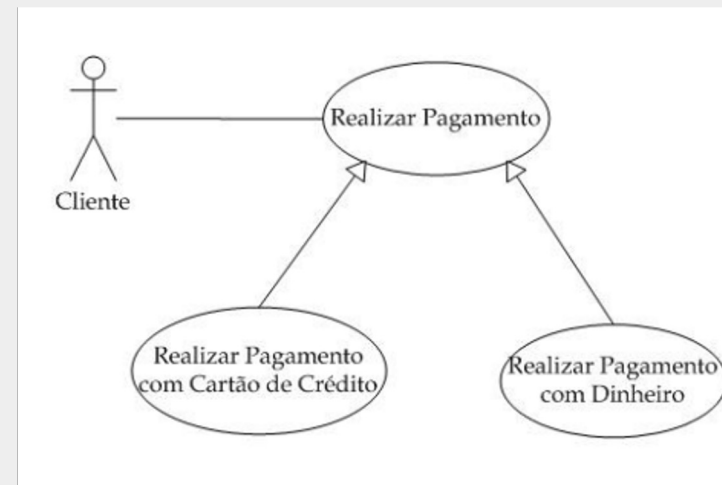
Fonte: Bezerra (2007)



# Relacionamento de generalização.

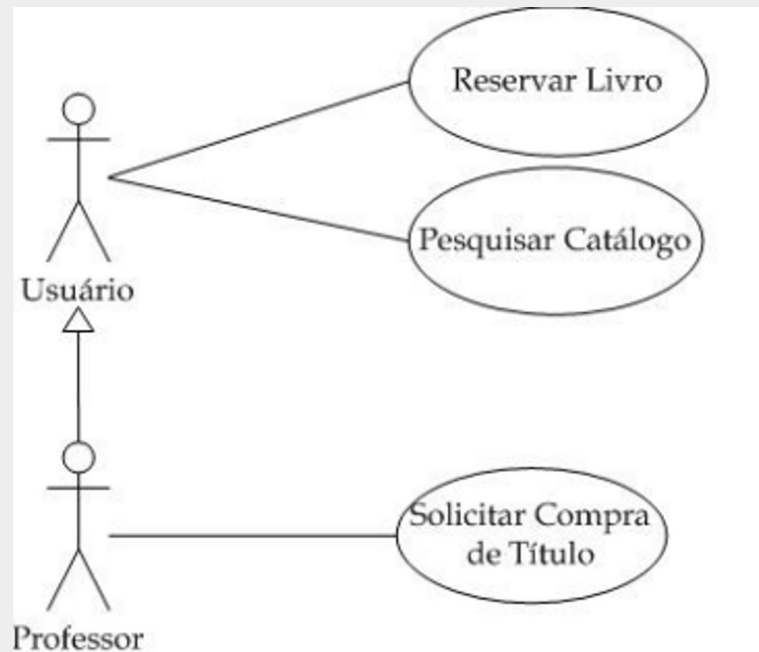
Na modelagem UML, um relacionamento de generalização é aquele no qual um elemento de modelo (o filho) tem como base outro elemento de modelo (o pai). Os relacionamentos de generalização são utilizados em diagramas de classe, componente, implementação e caso de uso para indicar que o filho recebe todos os atributos, operações e relacionamentos definidos no pai.

**Realizar Pagamento com Cartão de Crédito e Realizar Pagamento com Dinheiro herdam características em relação ao caso de uso Realizar Pagamento, especializando o seu comportamento.**

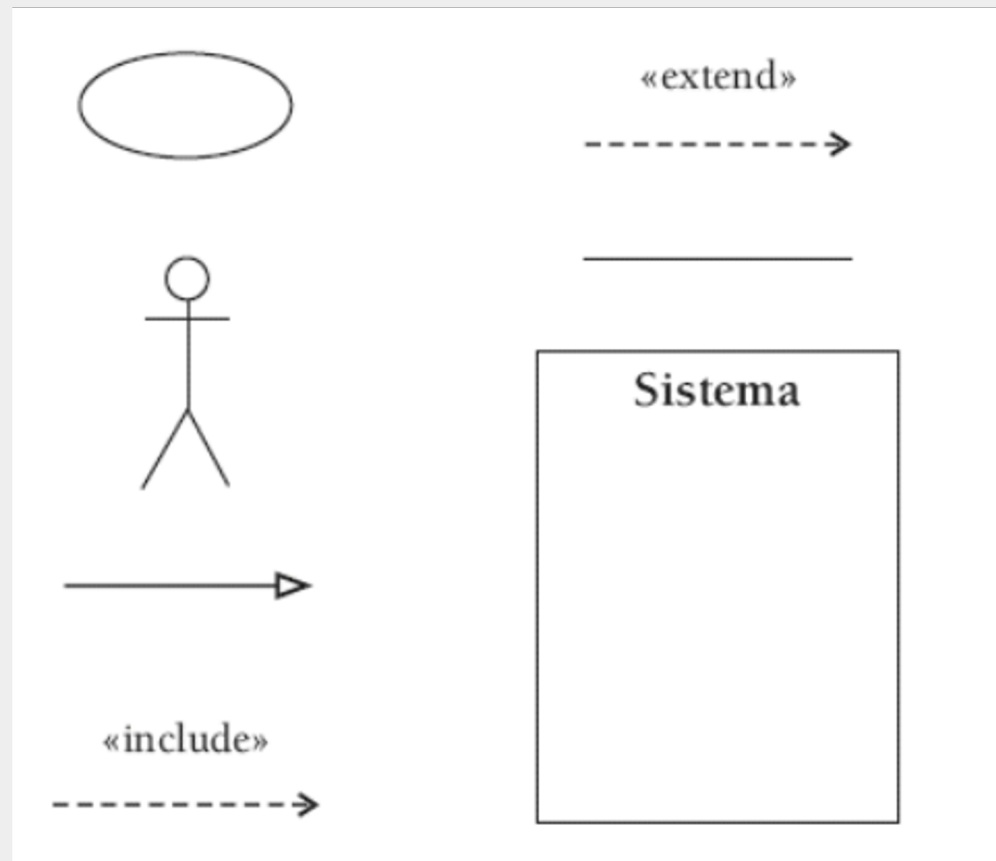


# Relacionamento de generalização.

A generalização entre os atores Usuário e Professor indica que este último pode interagir com qualquer caso de uso que um usuário comum interage.



# Resumo das Notações



# Vamos Praticar?

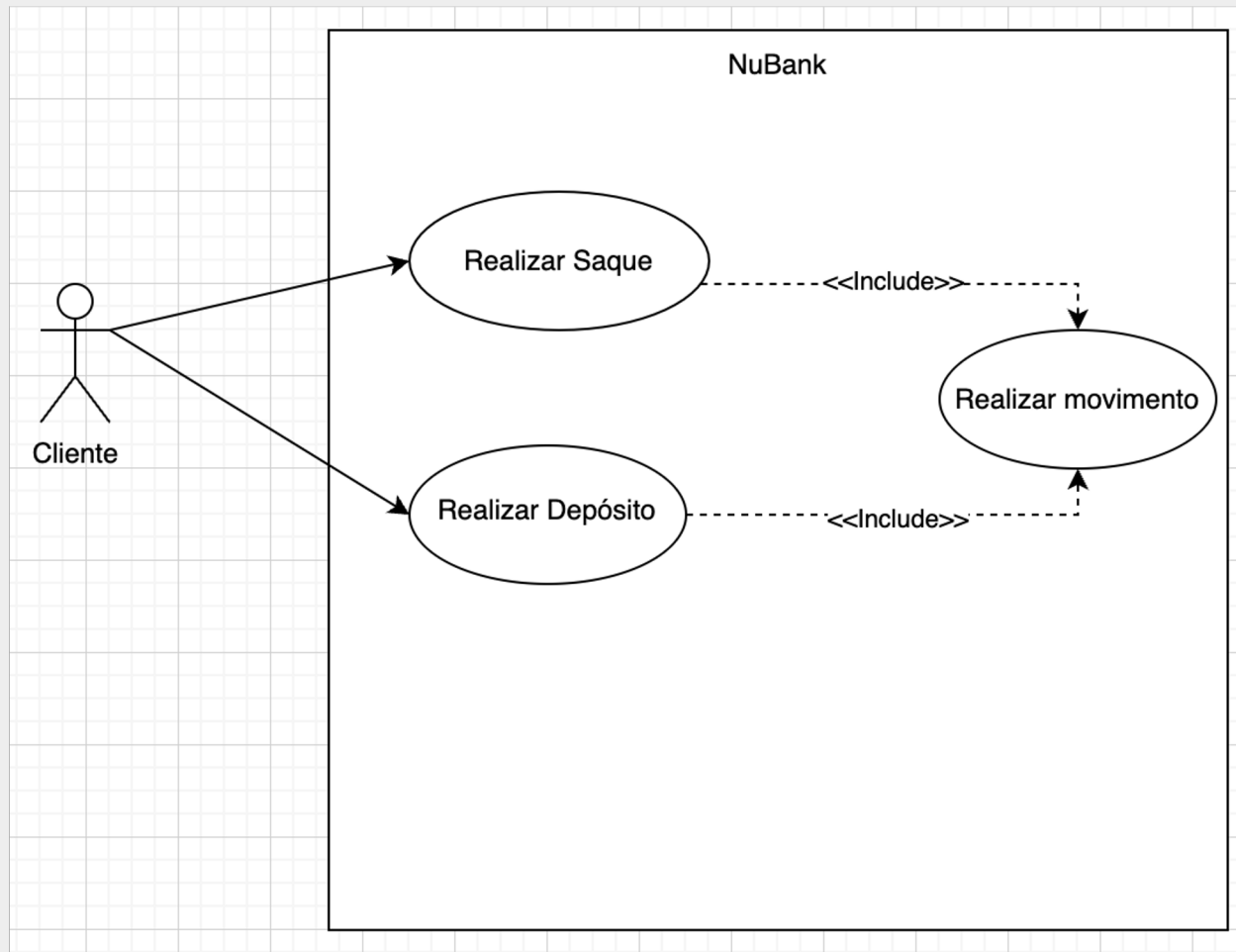
**Considerando o seguinte cenário:**

**Um cliente vai até o banco realizar um saque ou um depósito e ambas as operações devem registrar movimentos.**

**<https://online.visual-paradigm.com/>**



# Vamos Praticar?



**Além dos conceitos e dos elementos do MCU, tem mais alguma coisa que precisamos conhecer?**



**Tão importante quanto dominarmos a notação do MCU é termos conhecimento de técnicas e boas práticas de modelagem.**

**Se seguidas e utilizadas, nos levam à construção de modelos coerentes com as reais necessidades dos futuros usuários.**

**Atores e os casos de uso são identificados a partir de informações coletadas no levantamento de requisitos (analistas devem identificar as atividades do negócio relevantes ao sistema a ser construído).**

**Não há uma regra geral que indique quantos casos de uso e atores são necessários para descrever um sistema.**

**A quantidade de casos de uso e atores depende da complexidade do sistema.**



**“Um ator é todo elemento externo que interage com o sistema.”**

**Atores em potencial: Fontes e os destinos das informações a serem processadas.**

**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.**

## **Perguntas úteis:**

- » Que órgãos, empresas ou pessoas (cargos) irão utilizar o sistema?**
- » Que outros sistemas irão se comunicar com o sistema?**
- » Alguém deve ser informado de alguma ocorrência no sistema?**
- » Quem está interessado em um certo requisito funcional do sistema?**

**Devemos continuar pensando sobre atores quando passar para a identificação dos casos de uso.**

A partir da lista (inicial) de atores, deve-se passar à **identificação dos casos de uso**.

Nessa identificação, pode-se distinguir entre dois tipos de casos de uso:

✓ **Primário:** representa os objetivos dos atores.

✓ **Secundário:** aquele que não traz benefício direto para os atores, mas que é necessário para que sistema funcione adequadamente.

## **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?**

**» Caso de uso “oposto”: Deve-se perguntar para cada caso de uso: “As ações realizadas pelo sistema quando da realização deste caso de uso podem ser desfeitas?”.**

**» Caso de uso que precede outro caso de uso: Por exemplo, para que um cliente realize uma compra, é necessário que ele esteja cadastrado no sistema. A pergunta a ser feita para cada caso de uso, “o que pode ocorrer antes da realização deste caso de uso?”.**

**» Caso de uso que sucede a outro caso de uso: A pergunta geral que se deve é “o que pode ocorrer após a realização deste caso de uso?”.**

**» Caso de uso temporal: Por exemplo: “O sistema deve gerar um relatório de vendas toda sexta-feira”. A pergunta geral nessa situação é: “Há alguma tarefa que o sistema deva realizar automaticamente?”.**

**» Caso de uso relacionado a alguma condição interna:**

**Seguem dois exemplos disso: “O sistema deve notificar o usuário de que há novas mensagens de correio”;**

**“O sistema deve avisar o almoxarife de que um determinado produto chegou no nível de estoque mínimo”.**

## **Categorias:**

- ✓ Manutenção de cadastros;**
- ✓ Manutenção de usuários;**
- ✓ Gerenciamento de acesso;**
- ✓ Manutenção de informações provenientes de outros sistemas.**

**O sistema de software não existe para cadastrar informações, nem tampouco para gerenciar os usuários.**

**O objetivo principal de um sistema é agregar valor ao ambiente no qual ele está implantado.**



# Inclusões, Alterações, Exclusões e Consultas são casos de uso?

Para cada objeto do sistema, seria necessário considerar três casos de uso?

**Exemplo:**

**Incluir Livro, Alterar dados do livro e Excluir Livro**

**E quanto às consultas, por exemplo, Consultar Livro por Autor, Consultar Livro por Título, etc.,**

**Seriam casos de uso?**

# Inclusões, Alterações, Exclusões e Consultas são casos de uso?

Não há consenso sobre isso.

Como são processos em geral muito simples, com lógica conhecida, não vamos detalhá-los como faremos com os casos de uso mais complexos.

No entanto, incluiremos no diagrama de casos de uso, para dar uma noção geral do escopo e tamanho do sistema.

Ex:

Manter Usuários, Manter Clientes.

