

Neste artigo vou falar um pouco sobre modelagem de sistemas usando **UML** focando exclusivamente os **diagramas de casos de uso**.

A primeira coisa que devemos ter em mente é que os princípios aqui discutidos não se referem a uma linguagem específica; estamos focando é claro a análise orientada a objetos onde conceitos como encapsulamento de atributos e métodos, alta coesão e baixo acoplamento, herança e polimorfismo devem estar bem assimilados.

Vamos usar a **linguagem UML** que é um modelo de linguagem que define uma notação que são todos os elementos de representação gráfica vistos no modelo.

Estamos pois na fase de análise e não estamos preocupados com software nem hardware.

## Caso de Uso - definições

Segundo Ivan Jacobson, podemos dizer que um caso de uso é um *"documento narrativo que descreve a sequência de eventos de um ator que usa um sistema para completar um processo"*.

Um caso de uso é uma técnica de modelagem usada para descrever o que um novo sistema deve fazer. Ele é construído através de um processo iterativo no qual as discussões entre o cliente e os desenvolvedores do sistema conduzem a uma especificação do sistema da qual todos estão de acordo.

Um caso de uso descreve as operações que o sistema deve cumprir para cada usuário. Ele vai ajudar a formalizar as funções que o sistema precisa fazer. Um *caso de uso* se apresenta como uma lista completa das interações entre um usuário e o sistema para cumprir uma tarefa. Lista completa significa que o caso de uso descreve as interações desde o início da tarefa, até o fim.

Casos de uso têm que ser compreensíveis por usuários por que só eles sabem o que o sistema precisa fazer. Os casos de uso permitem verificar se o desenvolvedor e o usuário concordam sobre o que o sistema deve fazer. Isso é um problema importante no desenvolvimento de software. No mesmo tempo, casos de uso podem servir de **"contratos"** entre os usuários e a equipe de desenvolvimento.

Os casos de uso tem por objetivo:

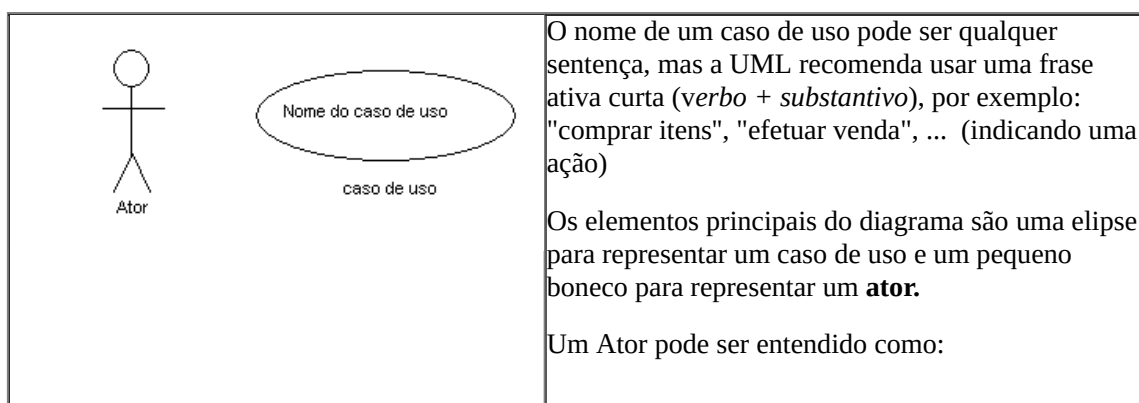
- Decidir e descrever os requisitos funcionais do sistema.;
- Fornecer uma descrição clara e consistente do que o sistema deve fazer;(Descrevem O QUE fazer e não COMO fazer)
- Permitir descobrir os requisitos funcionais das classes e operações do sistema. (Casos de uso NÃO são requisitos)
- Documentar e entender os requisitos de um sistema;
- Delimitar o contexto de um sistema;

Desta forma os casos de uso funcionam como uma fase de transição entre os requisitos e a análise.

Podemos dizer que os componentes de um modelo de casos de uso são:

- **Ator** - é um papel que tipicamente estimula/solicita ações/eventos do sistema e recebe reações. Cada ator pode participar de vários casos de uso
- **Casos de uso** - documento narrativo que descreve a sequência de eventos feitos por um ator no uso do sistema.
- **Sistema** - O sistema a ser modelado

Na **UML** o modelo de casos de uso consiste de diagramas de casos de uso que mostram os atores, os casos de uso e seus relacionamentos. Os elementos gráficos que representam atores, casos de uso e sistema são mostrados abaixo:



	<ul style="list-style-type: none"> <li>- Uma entidade externa ao sistema que participa de um ou mais casos de uso.</li> <li>- Fornecem eventos de entrada e recebem alguma resposta do sistema;</li> <li>- Podem ser: Papéis que pessoas representam nos casos de uso , dispositivos de hardware , outros sistemas, tempo;</li> </ul>
--	---

**Nota: Abaixo temos uma estrutura de especificação que você pode usar para casos de uso. Não existe um padrão.**

#### **Nome do caso de uso**

Descrição do caso de uso (um parágrafo).

#### **Atores**

Lista dos nomes dos atores com descrição curta.

#### **Prioridade**

Seja este caso de uso muito importante no projeto ou acessório?

#### **Pre-Condições**

Lista de condições que têm que ser verificadas antes que o caso de uso começa.

#### **Fluxo de eventos**

##### **Fluxo principal**

1. Primeiro passo no caso de uso.
2. Segundo passo no caso de uso.
3. ...

##### **Fluxos alternativos**

Descrever os fluxos alternativos.

##### **Pós-Condições**

Lista de condições que têm que ser verificadas depois do fim do caso de uso.

##### **Pontos de extensão**

Lista dos pontos de extensão no caso de uso.

##### **Casos de uso incluídos**

Lista dos nomes dos casos de usos incluídos.

Nos primeiros contatos com os modelos de casos de uso surgem com frequência três perguntas para as quais não existe uma resposta absoluta , são elas ::

#### **1 - Como identificar atores ?**

Para identificar os atores que vão participar do modelo devemos fazer as seguintes perguntas :

- Quem usa o sistema ?
- Quem inicia o sistema ?
- Quem fornece os dados ?
- Quem usa as informações ?

#### **2- Como descrever atores ?**

Geralmente descreve atores usando :

- Nome do caso de uso
- tipo de uso (frequente, ocasional , etc...)
- descrição de seu papel no sistema

#### **3- Como Identificar casos de uso ?**

Os casos de uso são interações entre os atores e o sistema . Temos então ações do ator e ações do sistema. Sendo que os atores sempre iniciam a ação.

Vamos dar um exemplo prático para que tudo fique mais claro. Vamos supor , por questão de simplicidade , que temos que modelar usando casos de uso a compra de item em um a loja com um terminal de ponto de venda.

### Quais são os atores ?

Quem usa o sistema é o **cliente** e ele usa um **terminal de caixa**.

### Como podemos identificar o caso de uso ?

Podemos chamar este caso de uso de : **Comprar Item** (*verbo+substantivo*)

Agora vamos a uma descrição textual do caso de uso **Comprar Item** onde atuam os atores **cliente e caixa**. (*Aqui estou adotando uma estrutura de especificação bem simplificada por questões didáticas*)

#### Caso de uso - Comprar Item

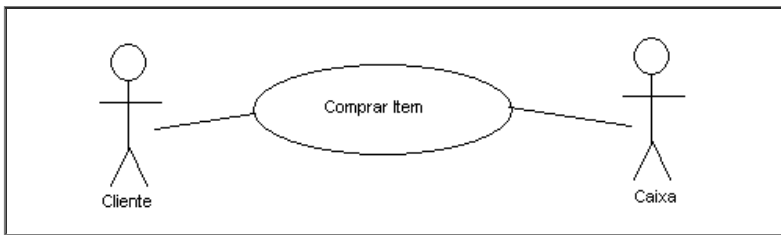
**Atores - Cliente , Caixa**

**Descrição - Este caso de uso começa quando um cliente chega ao terminal com itens que deseja comprar.**

**O caixa registra os itens , recebe o pagamento e emite uma nota fiscal.**

**O Cliente recebe os itens comprados.**

Na UML temos o diagrama de caso de uso que pode ser representado para o caso acima da seguinte forma:



Algumas considerações :

- Nomeie um caso de uso começando com um verbo , para enfatizar que ele é um processo. *Ex: Cadastrar Cliente , Comprar Item , etc.*

- Para identificar claramente um ator iniciador e um evento , comece a descrição da sequência de um caso de uso usando o seguinte esquema:

- Este caso de uso começa quando o **<Ator> <Evento que inicia o caso de uso>**

Ex: Este caso de uso começa quando um cliente chega com vários itens para comprar

Vamos a um outro exemplo:

Suponha que você tenha um almoxarifado de peças onde clientes façam pedido e onde um operador receba tarefas do sistema para buscar peças para os pedidos dos clientes e distribuir peças do setor de compras para o almoxarifado. (O exemplo é bem simples para facilitar o entendimento do conceito)

Como poderíamos identificar os atores e os casos de uso para este exemplo?

Vamos identificar os atores . Eles são : **Cliente , Operador , Sistema e Setor de Compras.**

Certo ?

**Não errado !!!** No exemplo acima **Sistema** não pode ser um ator pois ele não se ajusta ao conceito dado a um ator : **Um agente externo ao sistema.**

Lembre-se que um ator é um papel que interage com o sistema mas não faz parte do sistema, então no lugar de **Sistema** poderíamos sugerir um administrador ou gerente. Então os atores seriam : **Cliente , Operador , Administrador e Compras.** (Podem existir sub-sistemas que interagem com o sistema , neste caso eles seriam considerados atores.)

E os casos de usos , quais seriam ?

- solicitar peças (ator Cliente)
- entregar peças (ator Compras)
- buscar pedidos (ator operador)
- distribuir pedidos (ator operador)
- cadastrar Tarefas (administrador)

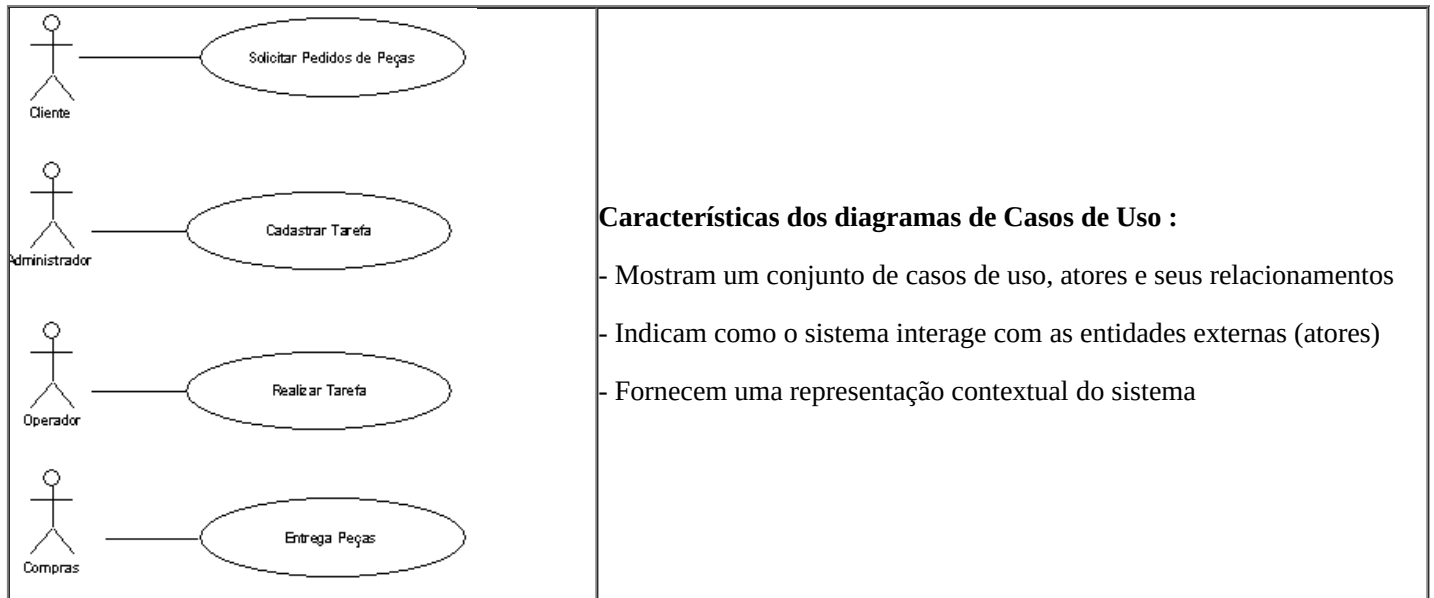
Certo ?

**Errado !!!** No caso do ator operador ele não atua sobre o sistema nos casos de uso **buscar pedidos** e **distribuir pedidos**, ele atua sobre o sistema realizando uma Tarefa.

Então o correto seria.

- solicitar peças (ator Cliente)
- entregar peças (ator Compras)
- realizar Tarefa (ator operador)
- cadastrar Tarefas (administrador)

Usando a representação UML para os **diagramas de casos de uso** teríamos :



Este seria o caso de uso preliminar(simplificado) pois não temos muito detalhamento nesta etapa do modelo. A próxima etapa seria realizar um refinamento do modelo a fim de obter o relacionamento entre os casos de uso através da **generalização, inclusão ou extensão**.

A partir do diagrama de casos de uso preliminar muitas vezes temos que definir casos de usos adicionais separadamente pois as operações se encontram duplicadas em outros casos de uso ou são complexas e longas e a separação nos ajuda a compreendê-las.

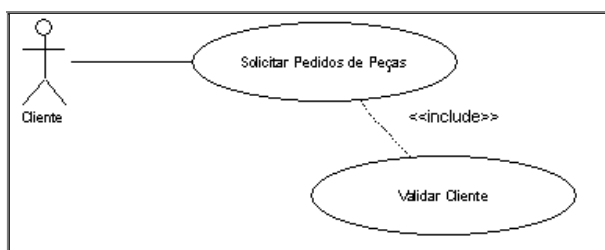
Os relacionamentos possíveis são :

**1- Inclusão :** Se um caso de uso inicia ou inclui o comportamento de outro, dizemos que ele usa o outro.

**Ex:** No caso de uso **Comprar Item** se o pagamento for feito com dinheiro podemos ter a inclusão **PagarComDinheiro**

O relacionamento de inclusão em UML é ilustrado com uma linha de generalização com o rótulo **<<include>>**.

Então para o exemplo do cliente com o use case Solicitar Pedidos de peças teríamos:

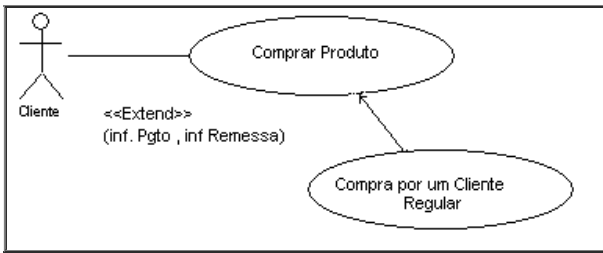


As propriedades básicas da inclusão são :

- realizar uma decomposição funcional
- reduzir a complexidade de um caso de uso
- O caso de uso básico não pode executar sem a inclusão.
- Comportamento comum

**2- Extensão** - Define pontos de extensão que adicionam comportamento a um caso de uso base descrevendo uma variação do comportamento normal. O caso de uso base pode ser executado mesmo sem a extensão.

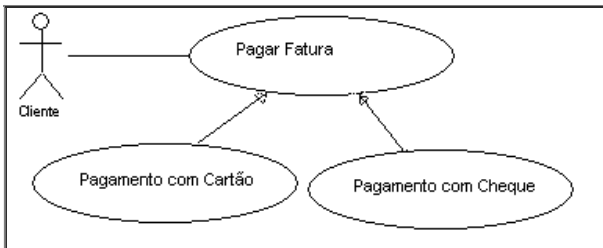
Ex: O caso de uso **Comprar Produto** pode apresentar a extensão **Compra por um Cliente Regular**. Abaixo temos o diagrama UML



Os pontos de extensão são indicados na linha entre os casos de uso do sistema.

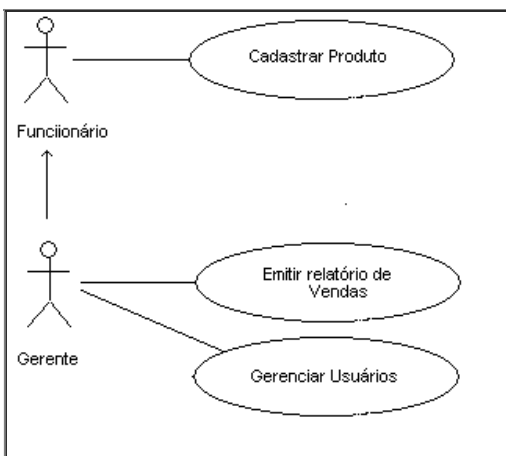
**3- Generalização** - Indica um caso de base que possui diferentes especializações e inclui comportamento ou sobrescreve o caso de uso base.

O caso de uso Pagar fatura apresenta as generalizações : **Pagamento com cartão** e **Pagamento com Cheque** , conforme o diagrama Abaixo:



Além disto temos também os relacionamentos entre atores onde um ator especializado pode acessar os casos de uso de um Ator base.

Abaixo temos um exemplo onde o Ator gerente acessa os casos de uso do ator funcionário



Espero que esta pequena introdução aos casos de uso amplie o seu horizonte para a modelagem **UML**.

Creio que uma das melhores ferramentas para modelagem é o **Rational Rose** , mas o preço é bem salgado. Como opção você pode usar uma das seguintes ferramentas :

- [Poseidon](#) - existe um versão open source (Community version)
- [Pacestar UML](#) - Uma versão shareware de uma ferramenta mais simples.
- [Rational Rose](#) - Trials e demos (**Rational Rose Limited Edition**).

Aguarde mais artigos onde irei falar sobre modelo conceitual , diagramas de sequência , diagramas de estado , e muito mais...

Até breve... 😊

[Veja os Destaques e novidades do SUPER DVD Visual Basic \(sempre atualizado\) : clique e confira !](#)

[Quer migrar para o VB .NET ?](#)

- Veja mais sistemas completos para a plataforma .NET no [Super DVD .NET](#) , confira...
- [Curso Básico VB .NET - Vídeo Aulas](#)

#### Quer aprender C# ??

- Chegou o [Super DVD C#](#) com exclusivo material de suporte e vídeo aulas com curso básico sobre C#.
- [Curso C# Basico - Video Aulas](#)

#### Quer aprender os conceitos da Programação

##### Orientada a objetos ?

- [Curso Fundamentos da Programação Orientada a Objetos com VB .NET](#) NEW

#### Quer aprender o gerar relatórios com o ReportViewer no VS 2013 ?

- [Curso - Gerando Relatórios com o ReportViewer no VS 2013 - Vídeo Aulas](#)

#### Quer aprender a criar aplicações Web Dinâmicas usando a ASP .NET MVC 5 ?

- [Curso ASP .NET MVC 5 - Vídeo Aulas](#)

Gostou ?  [Compartilhe no Facebook](#)  [Compartilhe no Twitter](#)

#### Referências:

- [Seção VB .NET do Site Macoratti.net](#)
- [Super DVD .NET - A sua porta de entrada na plataforma .NET](#)
- [Super DVD Vídeo Aulas - Vídeo Aula sobre VB .NET, ASP .NET e C#](#)
- [Seção C# do site Macoratti.net](#)
- [Super DVD C#](#)
- [Super DVD Visual Basic](#)
- [Curso Básico VB .NET - Vídeo Aulas](#)
- [Curso C# Básico - Vídeo Aulas](#)
- [ASP - Navegação com Site Map, Menu, TreeView e Master ...](#)
- [VB .NET - Populando o controle TreeView com tabelas e ...](#)
- [C # - Usando o controle TreeView - Macoratti.net](#)
- [Os novos controles de vinculação de dados - Macoratti.net](#)
- [Usando XmlDataSource - Macoratti.net](#)
- [Exibindo dados XML e filtrando com XPath - Macoratti.net](#)
- - [OMG](#) (Object Modeling Glossary).

- [Rational](#)
- Larman, Craig , "Utilizando UML e Padrões - Uma introdução à análise e ao projeto orientados a objetos" , Bookman, 2000

---

[José Carlos Macoratti](#)