

Administração Central
Cetec Capacitações

Conceitos de UML

Administração Central
Cetec Capacitações

1 Conceito de UML

A UML (Unified Modeling Language) que significa Linguagem de Modelagem Unificada é uma linguagem visual que serve para especificar, construir, visualizar e documentar sistemas orientados a objetos.

2 Conceito de Programação Orientada a Objetos

Na década de 60, Alan Key, criou o termo Programação Orientada a Objetos (POO), que se popularizou somente nos anos 90.

Mas o qual é o objetivo desse paradigma? O seu objetivo é tentar aproximar o mundo real do mundo virtual, utilizando os conceitos do mundo que temos a nossa volta na programação, ou seja, tudo é considerado um objeto e o mesmo possui várias funcionalidades.

A orientação a objetos não é exclusividade da linguagem ASP.net. Linguagens como C#, VB.net, PHP 5, Java e outras, fazem uso de POO.

Sendo assim, podemos classificar POO como uma técnica de desenvolvimento de softwares que busca representar os elementos do mundo real dentro de um produto de software, trazendo uma abordagem natural para os conceitos de programação de softwares.

Veja o exemplo abaixo:



No mundo real reconhecemos facilmente uma pessoa e suas características como: nome, idade, altura, peso, sexo e outras, e também conseguimos determinar quais ações este objeto pode ter, como andar.

Administração Central Cetec Capacitações

Em um programa reproduzimos as percepções do mundo real, com suas definições, características e ações. Uma pessoa tem: nome, idade, altura, peso e sexo e pode executar a ação andar.

Pessoa
- nome : String
- idade : int
- altura : int
- peso : double
- sexo : char
+ andar() : void

Resumindo a Orientação a Objetos é uma metodologia de programação que utiliza modelos organizados do mundo real para resolver os problemas nas linguagens de informação

E o “objeto”, a base dessa metodologia, é uma estrutura de dados com comportamentos de uma única entidade, ou seja, tudo o que podemos ver no mundo real é considerado um objeto com atributos e funcionalidades.

2.1 Classe e Objeto

Para entendermos os conceitos de classe e objeto, primeiramente precisamos entender o que é abstração. Abstração é a técnica ou habilidade de focar nos pontos principais de um determinado contexto, ignorando características menos importantes ou acidentais naquele contexto. Uma característica ignorada em um ponto pode ser relevante em outro momento e então será considerada.

Assim sendo, uma classe é apenas um modelo com foco nos aspectos principais para então serem criados objetos com suas características específicas, cada objeto criado terá sua própria forma.

Ainda está confuso? Vamos exemplificar

Objetos são elementos que pertencem ao mundo real, ou imaginário, e que podemos de alguma forma, identificar, como uma pessoa, uma caneta, uma carteira, entre outros.

Dentro do objeto são armazenadas informações em determinadas propriedades, que são chamadas, também de atributos. O objeto interage com

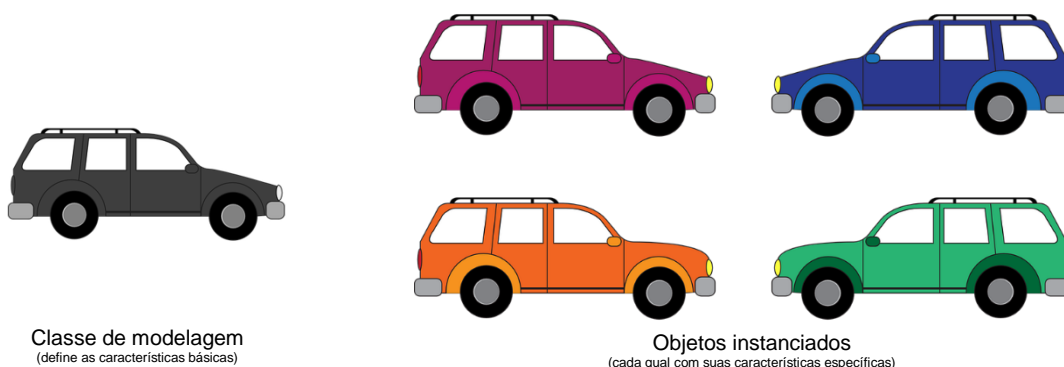
Administração Central Cetec Capacitações

o meio e, em função dos estímulos a que é exposto, realiza determinadas ações que alteram o estado e/ou informações de seus atributos.

A **classe** representa um conjunto de objetos com características semelhantes. Uma classe define o comportamento dos objetos por meio de métodos, definindo, os estados que ele é capaz de manter mediante atributos. Por exemplo, a classe “Pessoa” poderia ter os objetos “Fabiana”, “Ligia”, “Paulo”, “Marcos” e “Ana”. É possível criar quantos objetos forem necessários a partir de uma mesma classe, e todos terão as mesmas funcionalidades que foram definidas na classe.

Conclui-se que um objeto, por sua vez, é uma instância de uma classe.

Exemplo: A partir de um mesmo modelo podem ser construídos n veículos com as mesmas funcionalidades e características básicas, porém, cada carro construído terá suas próprias características específicas, como a cor, acessórios, motorização, etc.



Um objeto é uma entidade única capaz de reunir atributos e métodos, ou seja, ele reúne as propriedades dos objetos e as reações aos estímulos sofridas.

2.2 Atributos (ou Propriedades)

Todo objeto pode ter alguns atributos (propriedades) relacionados a ele. No exemplo anterior, em um carro, a cor do veículo, a cor dos para-lamas e o modelo de capô são os principais atributos. Cada objeto do tipo carro armazena uma informação diferente para cada um dos seus atributos.

Administração Central
Cetec Capacitações

2.3 Métodos (ou comportamentos)

Os objetos também podem ter determinados comportamentos relacionados a si. Os métodos são funcionalidades que poderão ser executadas pelo objeto, manipulando ou alterando os valores armazenados nos atributos

Como alguns exemplos, podemos citar os objetos e suas ações:

- Um carro tem o comportamento de acelerar;
- Um celular, o de fazer ligações;
- Um guarda-chuva, o de proteger da chuva.

3 Diagrama de Caso de Uso

Um caso de uso descreve uma sequência de ações do ponto de vista de um usuário com objetivo de representar o comportamento total ou parcial de um sistema estimulado pela interação do meio que se encontra.

Essa sequência é dividida em dois tipos de cenários:

Principal – é considerado o cenário perfeito, pois descreve uma sequência de ações que serão executadas sem nenhum problema durante a sua execução;

Alternativos – representa os cenários com exceções ou ações alternativas para execução de uma determinada ação.

Estes cenários representam os requisitos do sistema documentados na sequência que deverão ser executados.

Exemplo: Pedir uma pizza por telefone

Cenário Principal

1. O cliente realiza a ligação para a pizzaria.
2. A atendente anota o pedido.
3. A atendente confirma o endereço.
4. A atendente passa o valor total do pedido

No exemplo, temos um cenário perfeito, no qual nada ocorre de errado. Analisando esse cenário podemos modelar os seguintes cenários alternativos tratando algumas exceções

Administração Central
Cetec Capacitações

Alternativa: O cliente não sabe que sabor a pizzaria oferece

2a) O sistema disponibiliza para a atendente uma lista com as pizzas e seus respectivos valores e a mesma passa as opções para o cliente.

Alternativa: Endereço incorreto

3a) Solicitar ao cliente o novo cep, realizar a busca via sistema, confirmar o novo endereço e número com o cliente

Alternativa: Cliente precisa de troco

4a) O cliente informa o que será necessário troco de X

No exemplo as alternativas são apresentadas como subitens do cenário principal (2 – 2a; 3 – 3a; 4 – 4a). Esse procedimento facilita a associação da alternativa com o fluxo principal.

Na modelagem de um Caso de Uso é necessário definir as fronteiras do sistema, ou seja, definir quem ou o que será responsável em inserir e/ou receber informações processadas pelo sistema. Esse papel é representado por um ator, que pode ser uma pessoa, ou um grupo de usuários ou até mesmo um sistema. Fornecendo ao analista de sistema uma base para criar os perfis de acesso ao sistema

Representações

1 - Ator: representa um papel que o usuário vai desempenhar em relação ao sistema.



Exemplos:



Administração Central
Cetec Capacitações

2 – Caso de Uso: é uma descrição da ação que será executada.



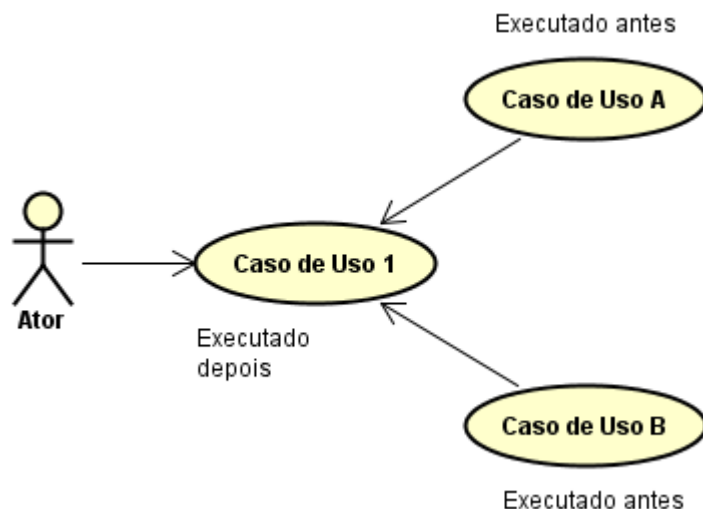
Exemplos:



Obs.: O nome do caso de uso pode ser colocado dentro ou abaixo dele.

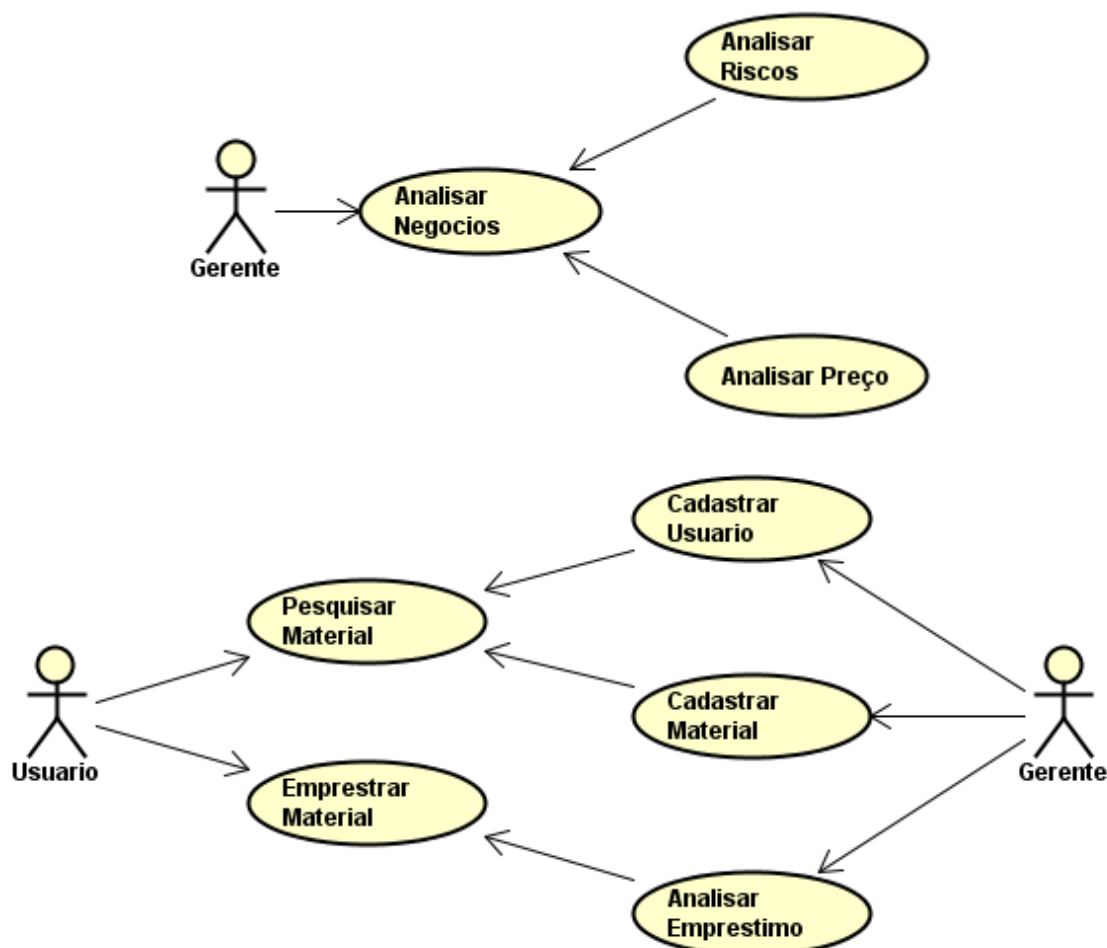
Dicas: Dar como nome aos casos de uso, frases verbais curtas no infinitivo (emprestar material) ou gerúndio (emprestando material). Pode ser utilizada qualquer uma dessas formas de nomenclatura, porém, deve-se prezar pela padronização, ou seja, se for escolhido uma das formas, deve-se utilizar em toda a modelagem.

3 – Associação de Extensão: é um tipo de associação que pode ser utilizado em diagramas de caso de uso. Neste relacionamento, a descrição de caso de uso pode ser estendida para outros casos de uso, através de associação de extensão.



Administração Central
Cetec Capacitações

Exemplos:



3.1 Tipos de Relacionamentos entre casos de uso e atores

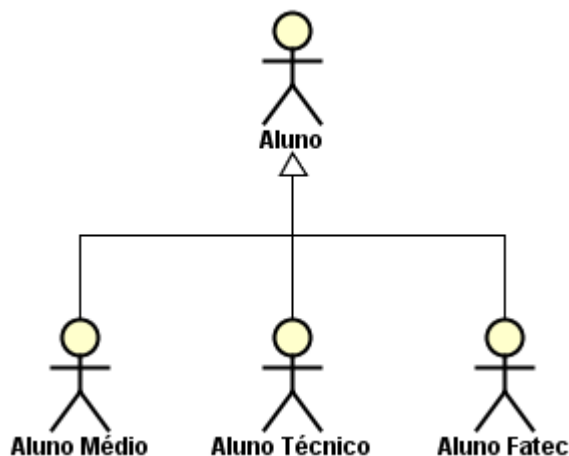
Os casos de uso são ações do sistema que não podem ser executadas sozinhas no contexto do sistema, sendo assim, os casos de uso precisam estar relacionados com outros casos de usos e/ou atores para enviar ou receber informações.

Associação – é a representação da comunicação do ator com o caso de uso para o envio ou recebimento de informações.

Generalização – esse tipo de relacionamento ocorre entre casos de uso ou entre atores, onde é aplicado o conceito de herança.

Administração Central
Cetec Capacitações

Exemplo:



3.2 Extensão (Extend)

O relacionamento de extensão ocorre entre casos de uso, demonstrando que um caso de uso terá uma ação adicionada, sendo assim, um ponto de extensão. Esses pontos de extensão são utilizados para representar as rotinas de exceção, normalmente para realizar o início de um cenário alternativo, pois o mesmo será executado apenas em uma determinada situação ou condição que seja satisfeita.

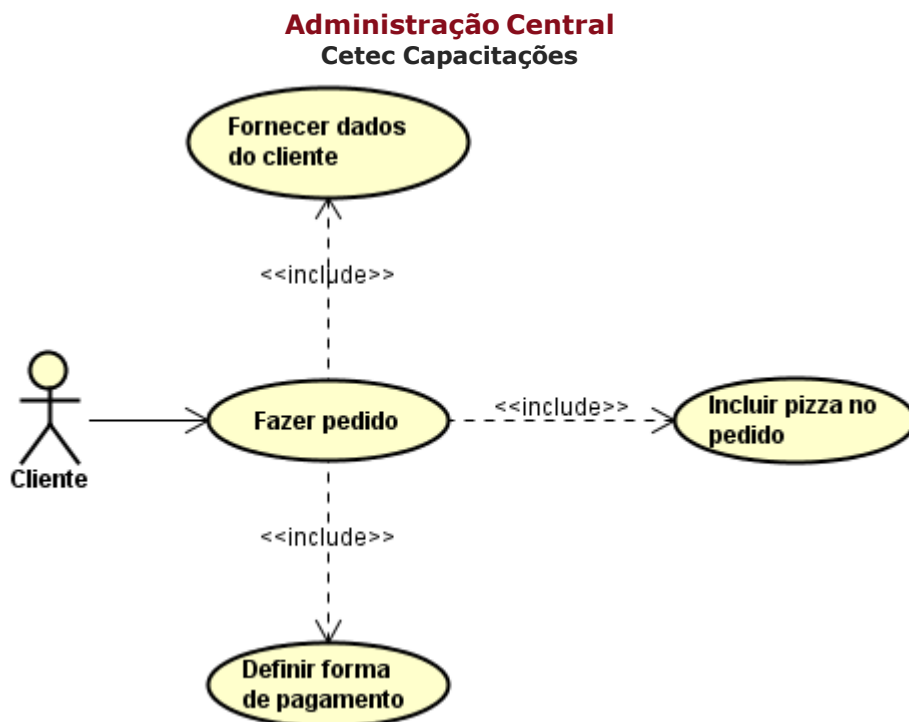
Exemplo:



3.3 Inclusão (Include)

É um relacionamento utilizado para realizar a cópia de um determinado caso de uso em outro caso de uso, proporcionando a reutilização do mesmo.

Exemplo:



No exemplo acima, os casos de uso Fornecer dados do cliente, Incluir pizza no pedido e Definir forma de pagamento foram copiados por dentro do caso de uso Fazer pedido, ou seja, ao executar o caso de uso Fazer pedido os demais também serão executados.

4 Diagrama de classe

O diagrama de classe auxilia na modelagem das classes do sistema de acordo com suas características e relacionamentos. Apesar das classes serem declaradas no diagrama de classes, as mesmas são utilizadas em outros diagramas da UML. A representação de uma classe é um retângulo subdividido em três compartimentos, separados por linhas horizontais que nesta ordem armazenam:

- O nome da classe;
- Os atributos;
- Os métodos

Os tipos de relacionamentos entre as classes são:

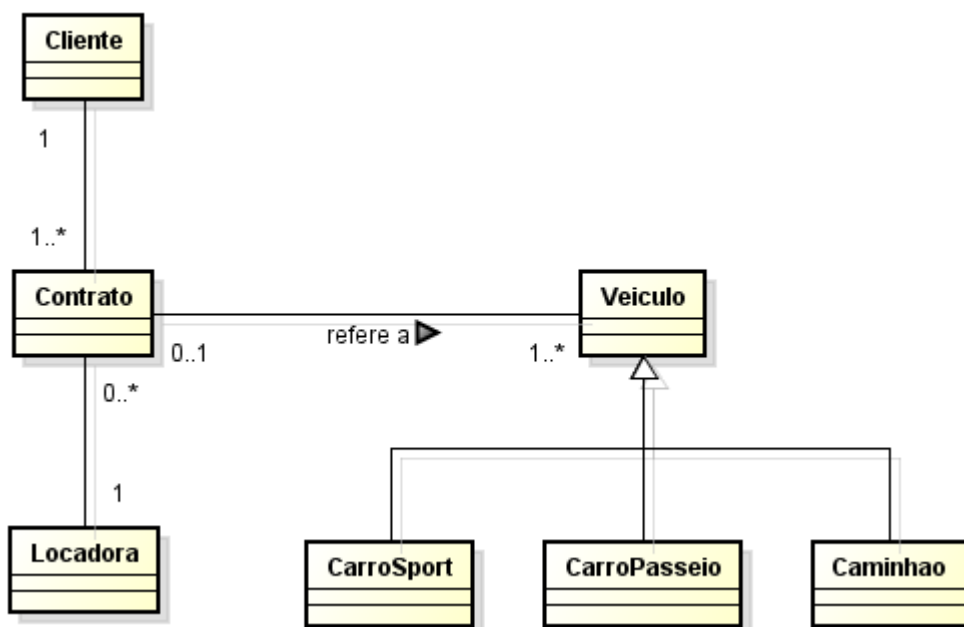
- associação (conectadas entre si),
- dependência (uma classe depende ou usa outra classe),
- especialização (uma classe é uma especialização de outra classe),
- entre outros que veremos a seguir

Administração Central Cetec Capacitações

Para representar um sistema normalmente são desenvolvidos vários diagramas de classe, já que não é possível inserir todas as classes em um único diagrama e as classes por sua vez podem estar presentes em vários diagramas de classe.

E como a estrutura apresentada em um diagrama de classe é válida em qualquer parte do ciclo de vida do sistema esse diagrama é considerado como estático.

Exemplo:



4.1 Visibilidade

Ao criar classes, atributos ou métodos precisamos definir qual é a sua visibilidade na aplicação. A visibilidade identifica por quem um atributo ou método pode ser utilizado. Como assim? Para ficar mais fácil lembre-se do conceito de definição de variáveis, pública, local, privada, etc, que define como as variáveis serão acessadas, o conceito é o mesmo, mas aqui trabalharemos com os seguintes tipos de visibilidade:

- **ou private** → visível apenas dentro da classe em que foi criado.
- + **ou public** → visível a partir de qualquer classe dentro da aplicação.
- # **ou protected** → visível na própria classe em que foi criado e em suas sub-classes, em breve estudaremos o conceito de classes filhas ou sub-classes.

Administração Central
Cetec Capacitações

~ ou package → visível dentro do mesmo pacote no qual a classe está declarada.

4.2 Atributos e Operações

Após identificar quais classes serão modeladas é necessário definir os atributos e/ou métodos que irão representar essas classes.

Um atributo representa uma característica que todos os objetos da classe deverão ter (por exemplo, todos ventiladores, têm uma cor, número de pás, tipo (teto, parede, piso, etc.), quantidade de velocidades, etc.). Mas cada objeto terá valores distintos para seus atributos (alguns ventiladores são azuis, outros são vermelho, etc.). O valor de um atributo pode ser alterado quando um método é executado, mas os atributos geralmente não mudam.

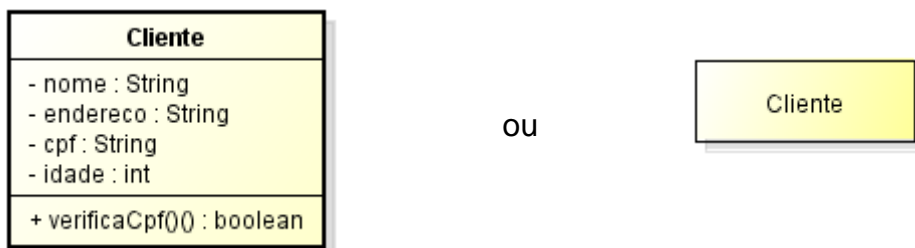
Os atributos dentro de uma mesma classe devem ter nomes diferentes, mas podem ter o mesmo nome se estiverem em classes diferentes. E o seu nome não pode conter caracteres especiais e nem iniciar com numeros, e quando o seu nome é formado por duas ou mais palavras deve-se capitalizar as primeiras letras das palavras, exceto da primeira palavras (ex: "nome"; "nomeCompleto").

Representações:

1 – Classe

É uma abstração de informações da realidade.

Exemplo:

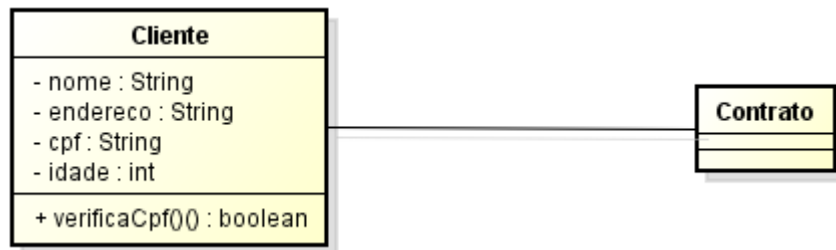


Administração Central
Cetec Capacitações

2 – Associação Simples

É um relacionamento ou conexão entre duas classes.

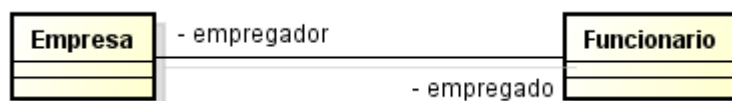
Exemplo:



3 – Papel

É o papel do relacionamento

Exemplo:



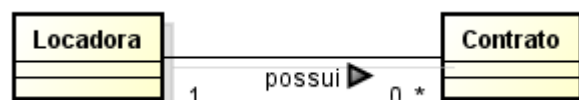
4 – Multiplicidade

- Exatamente uma – 1
- Vários (zero ou mais) – 0..*
- Opcional (zero ou uma) – 0..1
- Especificada Numericamente – n..m

5 – Direcionamento de Leitura

- ▶ Esquerda para direita
- ◀ Direita para esquerda

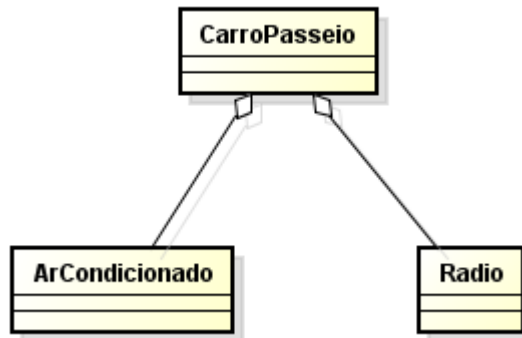
Exemplo:



Administração Central
Cetec Capacitações**6 – Agregação**

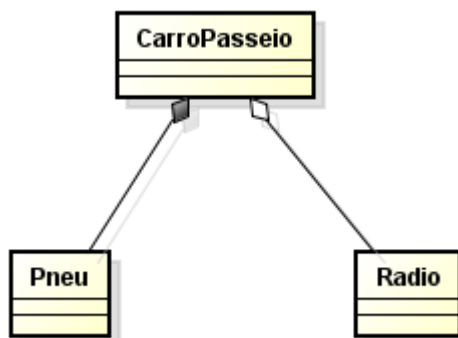
É um relacionamento de todo-parte, onde o todo não precisa das partes para existir.

Exemplo:

**7 – Composição**

É um relacionamento de todo-parte, onde o todo precisa das partes para existir.

Exemplo:



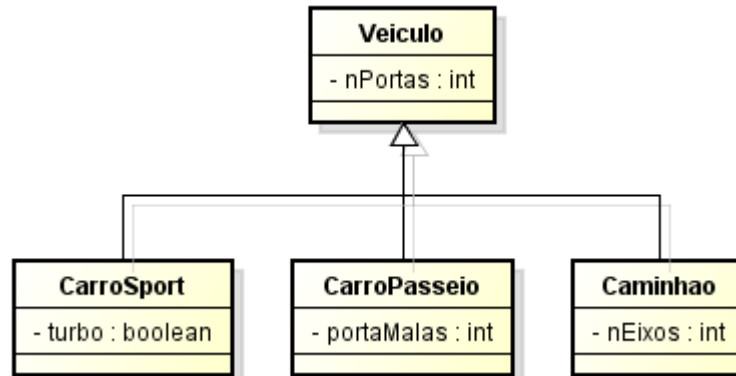
Obs.: A composição e a agregação pode ser demonstradas juntos.

8 – Generalização/Especialização

É um relacionamento onde as classes específicas herdam as características e ações da classe geral.

Administração Central Cetec Capacitações

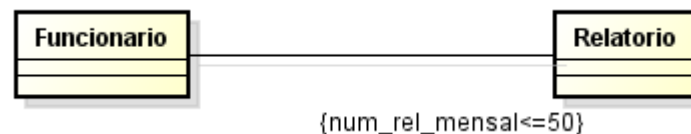
Exemplo:



9 – Restrições

Representa restrições relacionadas às classes.

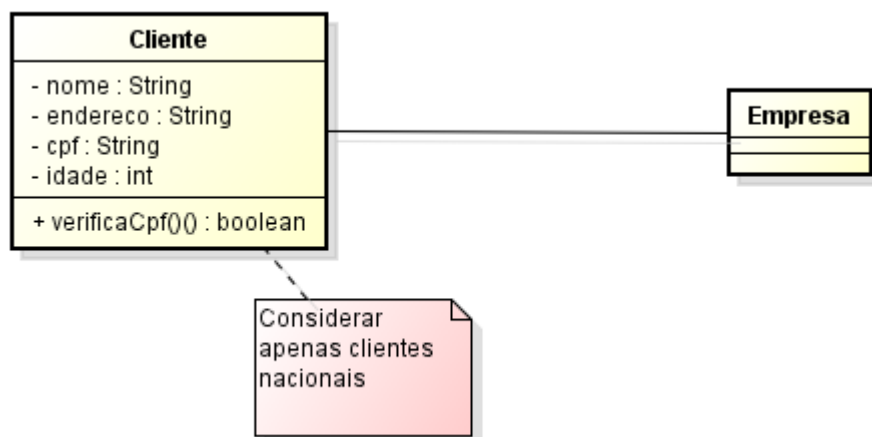
Exemplo:



10 – Nota

Trata-se de colocação de textos, informações e descrições importantes com relação às classes.

Exemplo:



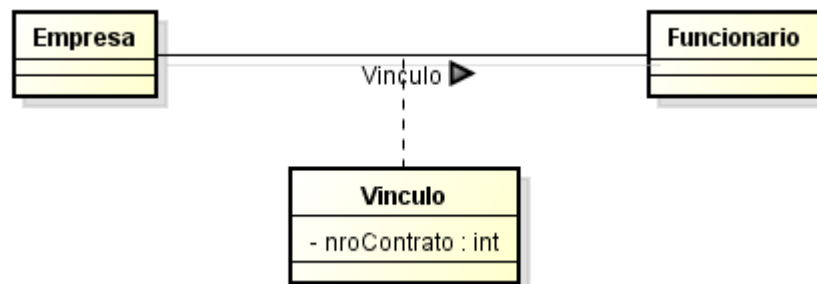
11 – Classe de associação

Trata-se de uma associação modelada com uma classe.

Administração Central Cetec Capacitações

A classe de associação só existe se existir o relacionamento entre duas classes que geram informações que não podem ser armazenadas nas mesmas.

Exemplo:



Exemplo geral: Diagrama de uma rede hospitalar.

