



**Guia para elaboração do Modelo de Domínio
Metodologia Celepar**

Agosto 2009

Sumário de Informações do Documento

Documento: guiaModelagemClassesDominio.odt

Número de páginas: 20

Versão	Data	Mudanças	Autor
1.0	15/10/08	criação	Danielle Mayer
1.0	21/07/08	Revisão e Alteração	Ariel, Danielle
1.0	28/07/08	Revisão e Alteração	Marcos, Danielle
1.0	12/06/09	Alteração	Marcos Chiarello
1.0	18/09/09	Revisão	Marcos, Danielle

Sumário

1	Introdução.....	4
1.1	Visão Geral	4
2	DIRETRIZES PARA MODELAGEM DE DOMÍNIO.....	5
2.1	Identificar Classes Conceituais e Atributos.....	5
2.1.1	Analisar Especificações de Casos de Uso	5
2.1.2	Identificar classes e atributos.....	6
2.1.2.1	Exemplo.....	7
2.1.3	Relatórios.....	10
2.2	Identificar Associações.....	10
2.2.1	Lista de Categorias de Associações.....	11
2.2.2	Exemplo.....	12
2.3	Refinamento do Modelo de Domínio.....	14
2.3.1	Generalização	14
2.3.1.1	Identificar Superclasses.....	15
2.3.1.2	Identificar Subclasses.....	15
2.3.1.3	Identificar Classes conceituais abstratas.....	16
2.3.1.4	Exemplo.....	16
2.3.2	Classes Associativas.....	17
2.3.2.1	Exemplo.....	18
2.3.3	Agregação e Composição	18
2.3.3.1	Exemplo.....	19
2.4	Organização do Modelo de Domínio.....	19
2.5	Considerações Finais.....	19

1 INTRODUÇÃO

A tarefa básica da análise orientada a objetos é identificar conceitos no domínio do problema e documentar os resultados em um Modelo de Domínio. O objetivo deste guia é orientar a atuação do Analista de Sistemas durante este trabalho.

1.1 Visão Geral

O Modelo de Domínio é a representação visual das classes conceituais ou objetos do mundo real em um domínio de problema, deve representar a compreensão da informação que o sistema vai gerenciar.

O Modelo de Domínio identifica os conceitos relacionados a requisitos do sistema e analisa o problema sob a perspectiva conceitual. É um artefato que representa o domínio do problema, portanto, não é utilizado para modelar a arquitetura de software(diagrama de classes de projeto), pois esta, embora inicialmente derivada do modelo conceitual pertence ao domínio da solução.

Assim, o Modelo de Domínio deve ser independente da solução física que virá a ser adotada e deve conter apenas elementos referentes ao domínio do problema em questão, ficando para a fase de projeto os elementos da solução, isto é, todos os conceitos que se referem a computadores como: interfaces, formas de armazenamento(banco de dados), segurança de acesso, comunicação, etc.

Usando a notação UML(*Unified Modeling Language*), um Modelo de Domínio é ilustrado com diagramas de classe, onde definições de operações ou responsabilidades são dispensadas.

Basicamente um Modelo de Domínio será composto por:

- Conceitos (classes conceituais);
- Atributos;
- Relacionamento entre classes conceituais.

2 DIRETRIZES PARA MODELAGEM DE DOMINIO

2.1 Identificar Classes Conceituais e Atributos

Esta atividade consiste em localizar classes conceituais e atributos relacionadas com os requisitos que estão sendo considerados, neste caso, os Casos de Uso envolvidos na iteração corrente deverão ser analisados.

O processo de identificação de classes conceituais e atributos é basicamente norteado pelas atividades:

- Analisar Especificações de Casos de Uso;
- Identificar Classes e Atributos.

2.1.1 Analisar Especificações de Casos de Uso

Analisar as Especificações de Casos de Uso identificando os substantivos ou expressões que denotam substantivos(conhecidas em lingüística como “sintagmas nominais”, ex: “item de locação”, “autorização de pagamento”, etc.) que serão considerados candidatos a classes conceituais ou atributos.

Cada substantivo identificado deve ser relacionado, agrupando palavras ou expressões que são sinônimos(como por exemplo “empréstimo” e “locação”).

Para auxiliar nesta análise observe a Tabela 1 abaixo listada.

Categoria	Significado
Entidades externas	<i>(outros sistemas, dispositivos e pessoas)</i> – que produzem ou consomem informação a ser usada pelo sistema.
Coisas	<i>(relatórios, figuras, cartas, sinais)</i> – que são parte do domínio de informação do problema.
Ocorrências de eventos	<i>(efetua pagamento, emite recibo)</i> – que ocorrem dentro do contexto da operação do sistema.
Papéis	<i>(gerente, engenheiro, vendedor)</i> – desempenhados por pessoas que interagem com o sistema.
Unidades organizacionais	<i>(divisão, grupo, equipe, setor)</i> – que são relevantes para o sistema.
Lugares	<i>(recepção, estoque)</i> – que estabelecem contexto do problema ou a função global do

Categoria	Significado
	sistema.
Estruturas	(sensores, impressora, computadores, leitora de código de barra) – que definem uma classe de objetos ou classes relacionadas de objetos.

Tabela 1: Categorização de Conceitos

Observação:

A correspondência do substantivo em uma das categorias é um indício de que ele é candidato a classe conceitual.

2.1.2 Identificar classes e atributos

Classes conceituais normalmente são identificadas a partir de conceitos complexos, que possuem um comportamento bem definido e não podem ser descritas por tipos alfanuméricos.

Atributos normalmente são identificados a partir de conceitos simples, que não tem comportamento definido e com um único tipo de dado associado. Identifique atributos no Modelo de Domínio, quando estes satisfazem a necessidade de memorizar informações referente aos requisitos(Casos de Uso) da iteração corrente.

Dentre os itens identificados, alguns serão classificados como **classes conceituais** (conceitos complexos), alguns serão classificados como **atributos** de classes e alguns poderão ser descartados como classes e atributos por serem irrelevantes para o sistema.

Para cada item verifique os seguintes questionamentos:

- O item fará parte do escopo do sistema? (ou é apenas uma informação dos agentes externos ao sistema)
- O item possuirá um comportamento identificado no sistema? (Possuirá responsabilidades/métodos)
- O item possuirá uma estrutura identificada no sistema? (estará associado a atributos)
- O item se relacionará a outro item?

Se a resposta para todos os questionamentos for sim, possivelmente o item identificado representa uma classe.

Durante a identificação de classes e atributos deve-se tomar atenção aos itens que possuem o mesmo significado. Se isso ocorrer, possivelmente eles representam o mesmo item.

2.1.2.1 Exemplo

Cenário principal do Caso de Uso **Processar Venda** de um sistema de Vendas

1. O **Cliente** chega ao ponto de **pagamento** com **bens** ou **serviços** para adquirir.
2. O **Caixa** inicia uma nova **venda**.
3. O **Caixa** digita o **Identificador do item**.
4. O Sistema registra a **linha de item de venda** e apresenta uma **descrição do Item**, o **preço** do mesmo e o **total** parcial da venda. O preço é calculado segundo um conjunto de regras de preços.
O Caixa repete os passos 3 e 4 até que indique ter terminado.
5. O Sistema apresenta o total.
6. O Caixa informa ao Cliente o total e solicita o **pagamento**
7. O Cliente paga e o Sistema processa o pagamento.
8. O Sistema registra a **venda** completada e envia as informações de venda e pagamento para o Sistema externo de **Contabilidade**(para contabilização e **comissões**) e de **Estoque**(para atualizar o estoque).
9. O Sistema apresenta o **recibo**.
10. O Cliente sai com o recibo e os bens(se for o caso)

Fluxos Alternativos

...

7a. Pagamento com dinheiro:

1. O Caixa digita a quantia de **dinheiro fornecida**.
2. O Sistema apresenta o **valor do troco** e libera a **gaveta de dinheiro**
3. O Caixa deposita o dinheiro fornecido e entrega o troco para o Cliente.
4. O Sistema registra o pagamento em dinheiro.

A partir da análise da Especificação do Caso de Uso Processar Venda foram relacionados itens candidatos a classes conceituais e atributos. Na sequência os mesmos foram classificados como apresentado na Tabela 2:

<i>Item Candidato</i>	<i>Nome da Classe ou Nome do Atributo</i>	<i>Classificação</i>
Cliente	Cliente	Classe
bens		Descartado como classe e Atributo Este descarte é possível porque bens e serviços, neste contexto, são sinônimos para item
serviços		Descartado como classe e Atributo Este descarte é possível porque bens e serviços, neste contexto, são sinônimos para item
identificador do item	Item	Classe

<i>Item Candidato</i>	<i>Nome da Classe ou Nome do Atributo</i>	<i>Classificação</i>
Caixa	Caixa	Descartada como classe e Atributo. Este descarte é possível porque não está definido nos requisitos a necessidade de ser conhecido ou de registro do Caixa atual. Claro que, se os requisitos mudarem, e surgir a necessidade de exibir no recibo a identificação do caixa este item candidato tornara-se-à uma Classe.
Venda	Venda	Classe
linha de item de venda	LinhaDeItemVenda	Classe
descrição do Item	descricao Para mostrar a descrição do item em tela ou recibo	Atributo
preço	preco Para calcular o total da venda e mostrar o preço da linha de item	Atributo
total		Descartado como classe e Atributo Descartado porque não é necessário memorizar esta informação(informação calculada)
pagamento	Pagamento	Classe
Contabilidade		Descartado como classe e Atributo Descartado porque não faz parte da iteração corrente.
comissões		Descartado como classe e Atributo Descartado porque não faz parte da iteração corrente.
Estoque		Descartado como classe e Atributo Descartado porque não faz parte da iteração corrente.
recibo		Descartado como classe e Atributo Descartado porque apenas representa um relatório da venda.
dinheiro fornecida (quantia)	quantia Para determinar se houve um pagamento suficiente e para calcular o troco.	Atributo
valor do troco		Descartado como classe e Atributo Descartado porque não é necessário memorizar esta informação(informação calculada)
gaveta de dinheiro		Descartado como classe e Atributo Descartado porque não é relevante para o sistema computacional.

Tabela 2: Exemplo de Itens Candidatos

Além dos candidatos extraídos do texto foram identificados outros conceitos não explícitos, conforme demonstrado na Tabela 3.

<i>Item Candidato</i>	<i>Nome da Classe ou Nome do Atributo</i>	<i>Classificação</i>
Loja	Loja Estabelecimento de Venda necessário para guardar dados necessários à emissão de recibos.	Classe
Especificação de Produto	EspecificacaoDeProduto Classe conceitual de especificação cujo objetivo é manter a descrição sobre o item, independente da existência física deste item.	Classe
quantidade	quantidade Para registrar a quantidade de um determinado item na venda	Atributo
endereço	endereco Para o caso do recibo requerer endereço e nome do estabelecimento da venda	Atributo ou Classe Obs: Neste exemplo consideraremos como atributo,
nome	nome Para o caso do recibo requerer endereço e nome do estabelecimento da venda	Atributo
data	data Um recibo é um relatório de uma venda que normalmente deve mostrar data e o horário da venda	Atributo
horário	Um recibo é um relatório de uma venda que normalmente deve mostrar data e o horário da venda	Atributo

Tabela 3: Exemplo de Itens Candidatos

Segue abaixo a Figura 1 que representa o primeiro esboço do Modelo de Domínio do exemplo:



Figura 1: Classes

2.1.3 Relatórios

Relatórios identificados deverão ser considerados candidatos a classes conceituais mediante a seguinte análise:

- Em geral, acrescentar um relatório em um modelo de domínio não é útil porque toda a sua informação é derivada de outras fontes, duplicando assim a informação que pode ser obtida de outra forma. Esta é uma razão para não considerá-lo;

Por exemplo: Um recibo é um relatório de venda.

- Agora se o relatório desempenha um papel especial em termos de regra de negócio. Esta é uma razão para considerá-lo.

Por exemplo:

- Um recibo confere ao portador o direito de devolver os itens comprados pelo mesmo.

2.2 Identificar Associações

Uma associação é um relacionamento entre classes que indica uma conexão (relação estática) com significado e interesse.

Para encontrar as associações entre classes conceituais deve-se observar cada classe e verificar se a informação representada está completa. Se não estiver, deve-se criar uma associação entre uma classe e outra a fim de complementar a informação necessária para que a classe

(conceito) faça sentido.

Por exemplo:

A classe Cliente, identificada no exemplo anterior, é uma informação aparentemente completa, que não necessita de associações para se complementar.

Já a classe Venda por si só não faz sentido, caso não se saiba qual cliente está relacionado ou qual(ais) itens de Venda (LinhaDeItemVenda) estão contidos na venda (o que está sendo vendido/comprado).

Outra maneira é verificar a necessidade de associações para as quais o conhecimento do relacionamento precisa ser preservado por algum tempo (associações que precisam ser conhecidas), que permitirão algum tipo de navegação no projeto e na implementação e podem ser úteis para facilitar o entendimento do modelo conceitual.

Por exemplo:

Um item de uma venda tem uma associação com a classe Venda para possibilitar o cálculo do total de vendas e a emissão de recibo.

Também é possível encontrar associações pelo uso da técnica de Lista de Categorias de Associação.

Observação:

1. Para nomear associações use uma frase com verbo que facilite a leitura quando juntada com os conceitos(classes) em cada extremidade.

Exemplo: Venda(classe) “**Paga-por**” Pagamento(classe)

2. **Direção:** Uma convenção útil é considerar a leitura de associações da esquerda para a direita ou de cima para baixo, embora a UML não faça disso uma regra.

2.2.1 Lista de Categorias de Associações

Técnica utilizada para encontrar associações tendo por base a lista de categorias de associações, conforme Tabela 4 que segue:

Categoria de Associações
A é uma parte física da B
A é uma parte lógica de B
A é fisicamente contido em B
A é logicamente contido em/sobre B
A é uma descrição de B
A é uma linha de item de uma transação ou relatório B
A é conhecido/registrado/relatado/captado por B
A é um membro de B
A é uma subunidade organizacional de B
A usa ou gerencia B
A se comunica com B
A está relacionado com uma transação B
A é uma transação relacionada com uma outra transação B
A é adjacente de B
A é possuído por B
A é um evento relacionado com B

Tabela 4: Categorias de Associação

Dentre as associações comuns as mais úteis em um Modelo de Domínio são:

- **A é uma parte física ou lógica de B**
- **A está fisicamente ou logicamente contido em B**
- **A é registrado por B**

2.2.2 Exemplo

Continuando o exemplo do Caso de Uso **Processar Venda** de um sistema de Vendas, as seguintes associações são identificadas, conforme Tabela 5:

Associação	Significado
Venda Paga-por Pagamento	Para saber se a venda foi paga, relacionar a quantia fornecida com o total da venda e imprimir o recibo A é uma transação relacionada com uma outra transação B(categoria de associação)
LinhaDeItemDeVenda Contido-em Venda	A é uma parte lógica de B (categoria de associação) A é uma linha de item de uma transação ou

Associação	Significado
	relatório B (categoria de associação)
EspecificaçãoDeProduto Descreve Item	A é uma descrição de B(categoria de associação)
Loja Registra-dados-da Venda	A é conhecido/registrado/relatado/captado por B Neste caso A representa a Venda e B representa a Loja
Venda Iniciada por Cliente	A é um evento relacionado com B (categoria de associação)
LinhaDeItemDeVenda Descrito por EspecificacaoDeProduto	Para saber qual produto é referente a cada linha de venda.
Loja Estoca Item	A está fisicamente contido em B Neste caso A representa a Venda e B representa a Loja

Tabela 5: Associações identificadas

Observações:

- Evitar associações redundantes ou deriváveis.
- Evitar associações que não indiquem necessidade de ser conhecidas de acordo com os requisitos, como demonstrado na Tabela 6:

Associação	Justificativa para descarte
Venda Iniciada - por Cliente	Os requisitos não indicam a necessidade de ser conhecido ou de registrar o cliente que dá início a uma venda.
Loja Estoca Item	Os requisitos não indicam a necessidade de ser conhecido ou de manter informações sobre o estoque. Os requisitos indicam que Estoque é um sistema externo.

Tabela 6: Associações descartadas

- Tendo em vista o quadro acima a associação **Venda iniciada por Cliente** não seria necessária, entretanto, ela é **importante** para a compreensão do domínio. Sendo assim, enfatize as associações que devem ser conhecidas, mas acrescente, conforme necessidade, associações destinadas somente a enriquecer a compreensão crítica do domínio.

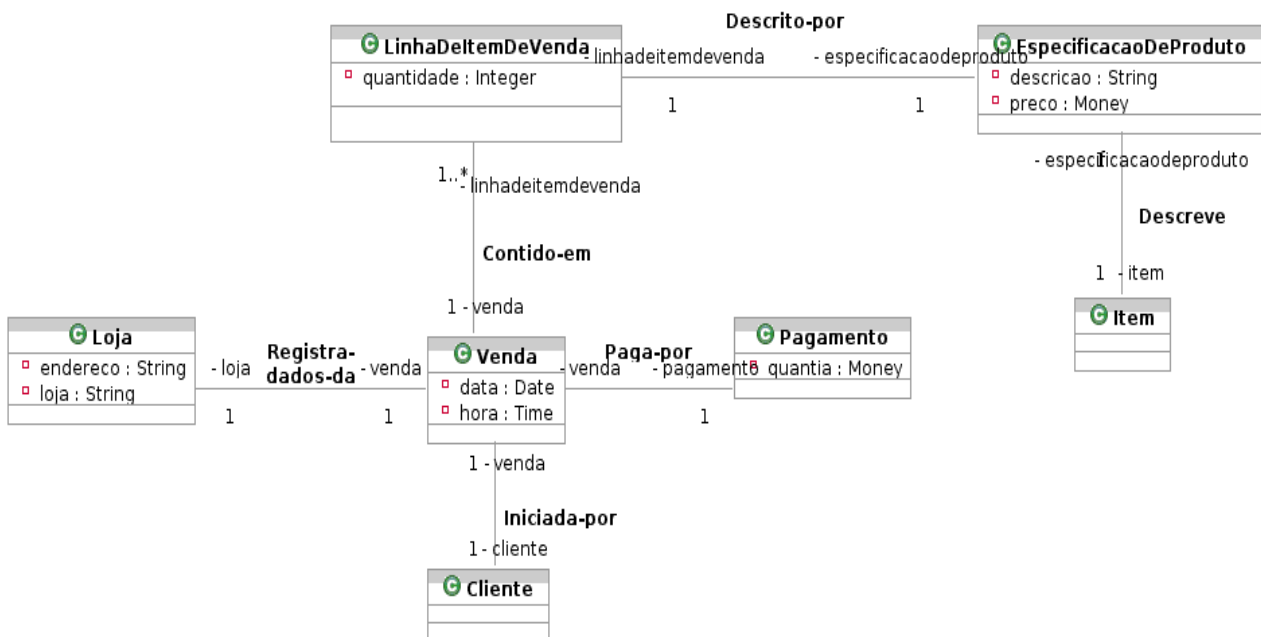


Figura 2: Modelo de Domínio com Atributos e Associações identificadas

2.3 Refinamento do Modelo de Domínio

Após a criação do Modelo de Domínio básico é importante o seu refino conforme as notações de Generalização, Classes associativa e Agregação/Composição.

2.3.1 Generalização

Generalização é a atividade de identificar o que há de comum entre os conceitos e definir relacionamentos entre superclasses (conceito geral) e subclasses (conceitos especializados). Trata-se de uma forma de construir classificações taxonômicas entre conceitos e ilustrá-las em hierarquia de classes.

Esta identificação é importante, pois permite o entendimento dos conceitos em termos mais gerais, refinados e abstratos. Conduz a uma compreensão aprimorada e a redução de informações repetidas, ou seja, a uma economia de expressão.

Passos para realizar a generalização da modelagem:

1. Identificar Superclasses;
2. Identificar Subclasses;
3. Identificar Classes Conceituais Abstratas.

2.3.1.1 Identificar Superclasses

A generalização em uma superclasse é aconselhada quando são identificados aspectos comuns entre subclasses em potencial.

Crie uma superclasse conceitual em um relacionamento de generalização com subclasses quando:

- As subclasses conceituais em potencial representarem variações de um conceito semelhante;
- As subclasses obedecem à Regra dos 100% e à Regra do É-um;
- Todas as subclasses tiverem o mesmo atributo, o qual possa ser decomposto e expresso na superclasse;
- Todas as subclasses tiveram a mesma associação, a qual possa ser decomposta e relacionada com a superclasse.

Regra dos 100% - > 100% da definição da superclasse conceitual deve ser aplicável à subclasse. A subclasse deve estar 100% de acordo com os atributos e associações da superclasse.

Regra É-um - > Todos os membros do conjunto de uma subclasse devem ser membros do conjunto de uma superclasse, ou seja, a Subclasse É uma Superclasse.

2.3.1.2 Identificar Subclasses

A fatoração de classes conceituais é a sua divisão em subclasses disjuntas. Sendo útil, este particionamento, quando:

- A subclasse tiver atributos adicionais de interesse;
- A subclasse tiver associações adicionais de interesse;
- O conceito da subclasse for operado, tratado, reagido ou manipulado de maneira

diferente da superclasse ou de outras subclasses, de forma que sejam interessantes a se considerar;

- O conceito da subclasse representa algo animado (por exemplo, um animal, um robô) que se comporta de maneira diferente da superclasse ou de outras subclasses.

Resumidamente, o principal motivo para extrair uma subclasse é a percepção de que uma classe tem comportamento usado por algumas instâncias da mesma (objetos) e não por outras .

2.3.1.3 Identificar Classes conceituais abstratas

A classe abstrata é sempre uma classe pai que não possui instâncias. Ela define um modelo para uma funcionalidade e fornece uma implementação incompleta - a parte genérica dessa funcionalidade - que é compartilhada por um grupo de classes derivadas. É útil identificar classes abstratas, pois elas restringem as classes para as quais é possível ter instâncias concretas, esclarecendo assim as regras do domínio do problema. [Uma classe abstrata está destinada apenas a servir como base para a criação de outras classes servindo como guia para a definição do comportamento dos herdeiros, das *subclasses*.]

Se cada membro de uma superclasse também deve ser membro de uma subclasse, então a superclasse é chamada **classe conceitual abstrata**.

A notação UML para representação é indicar seu nome em *itálico*.

2.3.1.4 Exemplo

Este exemplo representa que toda instância de *Pagamento* deve ser, mais especificamente, uma instância das subclasses *PagamentoEmDinheiro*, *PagamentoComCartaoDeCredito*, *PagamentoComCheque*.

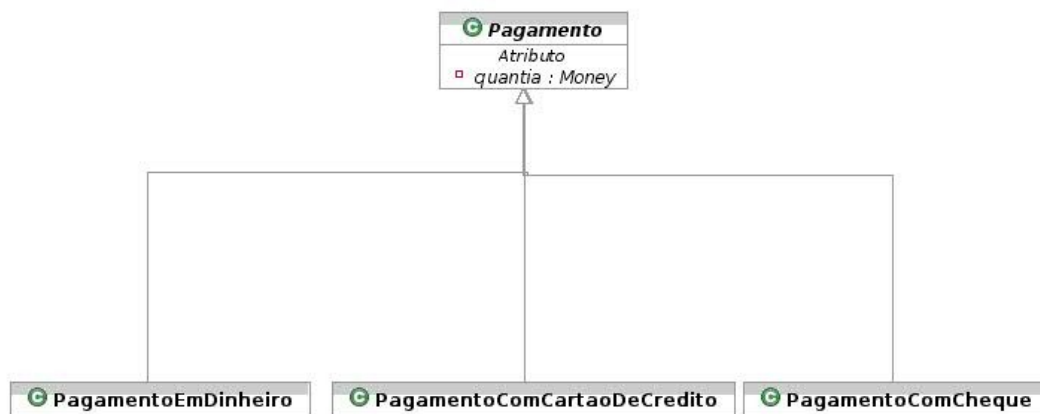


Figura 3: Generalização com classe conceitual abstrata

2.3.2 Classes Associativas

É uma classe que está ligada a uma associação, ao invés de estar ligada a outras classes. É normalmente necessária quando duas ou mais classes estão associadas, e é necessário manter informações sobre esta associação, podendo estar ligada a associações de qualquer tipo de conectividade.

A Classe de Associação pode ser vista como uma associação com propriedades de associação e de classe e devem ser utilizadas, quando necessário, para manter informações sobre uma associação específica entre objetos de duas classes relacionadas.

Classe Associativa pode ser útil em um Modelo de Domínio quando:

- Um atributo está relacionado a uma associação;
- As instâncias da classe associativa têm um tempo de vida dependente da associação;
- Há uma associação de muitos-para-muitos entre dois conceitos, bem como informações relacionadas à associação propriamente dita.

2.3.2.1 Exemplo

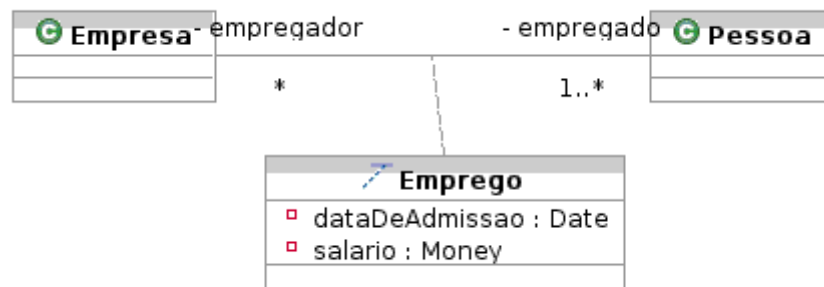


Figura 4: Classe Associativa

2.3.3 Agregação e Composição

Agregação é um tipo especial de Associação que demonstra que as informações de um objeto (**objeto-todo**) precisam ser complementadas pelas informações contidas em um ou mais objetos de outra Classe (**objeto-parte**).

Composição é um tipo de agregação onde o tempo de vida da **parte** é coincidente com o tempo de vida do **todo**. Neste caso, as partes só podem pertencer ao todo e são destruídas com ele.

Identificar e ilustrar agregações em um Modelo de Domínio **não é muito importante**, a menos que ela esclareça restrições existentes no domínio com relação à existência aceitável da parte independente do todo ou a parte pode não existir fora do tempo de vida do todo, no caso da composição. Sendo assim, a utilização de um relacionamento de agregação é subjetiva e não deve ser utilizada indiscriminadamente, utilize-a somente se o significado (semântica) efetivamente seja de especificar e documentar a indicação do “todo/parte” (no caso agregação) e, ainda, que “tempo de vida da parte é coincidente com o tempo de vida do todo (no caso composição).

Durante a modelagem da solução, classes de software, este tipo de associação (agregação/composição) será mais considerado devido ao impacto sobre as dependências criação-destruição existentes entre as classes de software, e sobre as classes persistentes que representam o todo e as partes (em termos de integridade referencial e caminhos de exclusão em cascata).

2.3.3.1 Exemplo

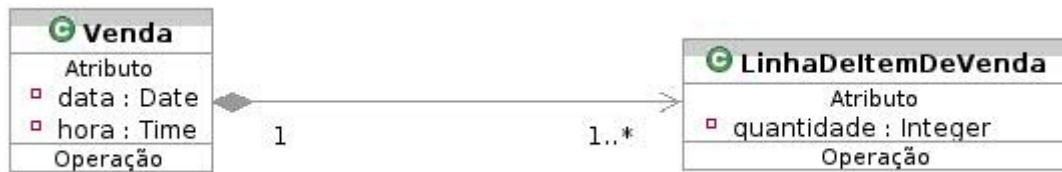


Figura 5: Classes com relacionamento de Composição

2.4 Organização do Modelo de Domínio

As classes do modelo de domínio deverão estar organizadas em um pacote, conforme notação da UML, denominado DOMÍNIO. Dentro deste pacote, se o modelo de domínio crescer muito, é interessante criar novos pacotes.

Para particionar o modelo de domínio em pacotes, agrupe elementos que:

- estejam ligados ao mesmo assunto - intimamente relacionados por conceitos ou finalidades;
- estejam juntos em uma hierarquia de classes;
- participem dos mesmos casos de uso;
- estejam fortemente associados.

2.5 Considerações Finais

Não existe Modelo de Domínio correto ou errado, todos os modelos são aproximações do domínio que estamos tentando compreender. Um bom Modelo de Domínio capta as abstrações essenciais e as informações necessárias para compreender o domínio no contexto dos requisitos considerados, auxiliando as pessoas na compreensão do domínio – seus conceitos, terminologia e relacionamentos.

Ao desenvolver um Modelo de Domínio, procure lembrar das seguintes diretrizes:

- É mais importante identificar classes conceituais do que associações;
- O excesso de associações tende a tornar o modelo de domínio confuso, em vez de esclarecê-lo;

- Evite mostrar associações redundantes ou deriváveis.