Esta página contém uma lista dos erros de revisão localizados no livro.

Se você identificou algum outro erro que não está listado aqui, por favor comunique ao autor pelo e-mail raul@inf.ufsc.br.

Última atualização em 29 de janeiro de 2007.

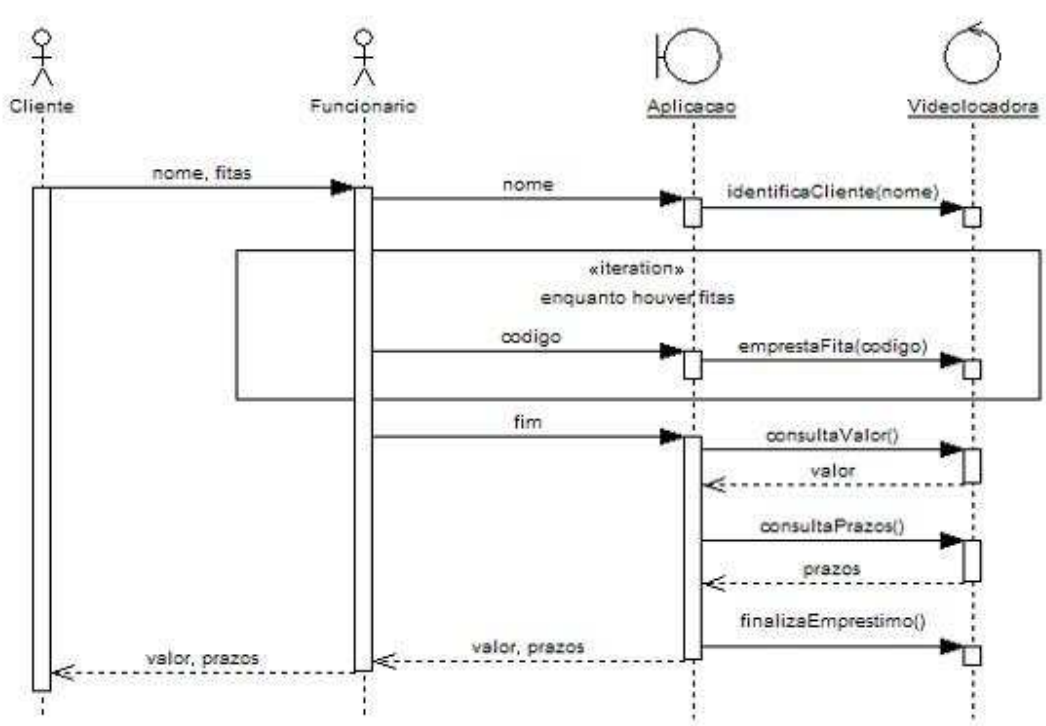
Capítulo 1 - Introdução.

Capítulo 2 - Concepção.

Capítulo 3 - Expansão dos Casos de Uso.

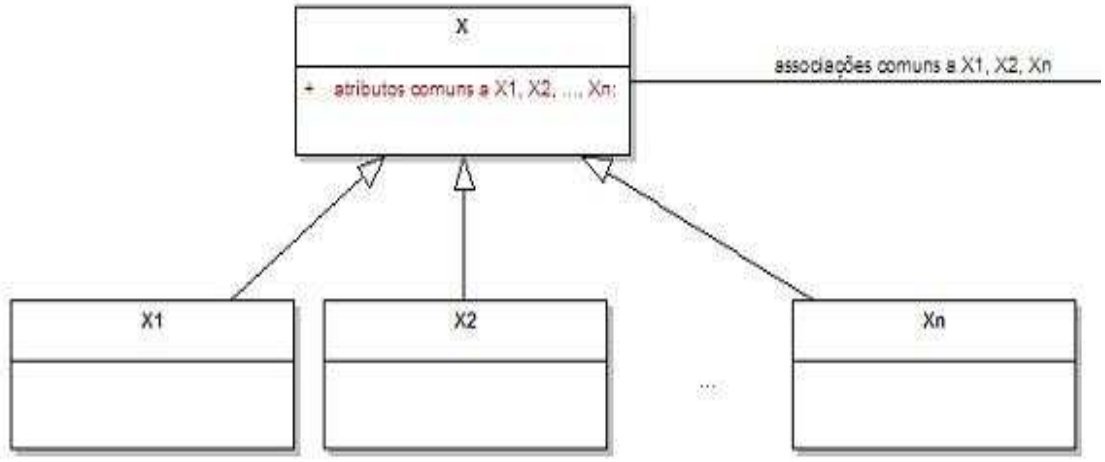
Capítulo 4 - Operações e Consultas de Sistema.

(08/03/2005): A figura 4.1 correta deveria ter uma seta sob "consultaValor()":



Capítulo 5 - Modelagem Conceitual.

(08/03/2005): A figura 5.23 correta deveria ser:



(08/03/2005): No capítulo 5, em várias figuras, a seta triangular de herança foi indevidamente substituída por uma seta simples que representa a associação direcional. Segue abaixo o conjunto de figuras onde este problema aconteceu com as devidas correções.

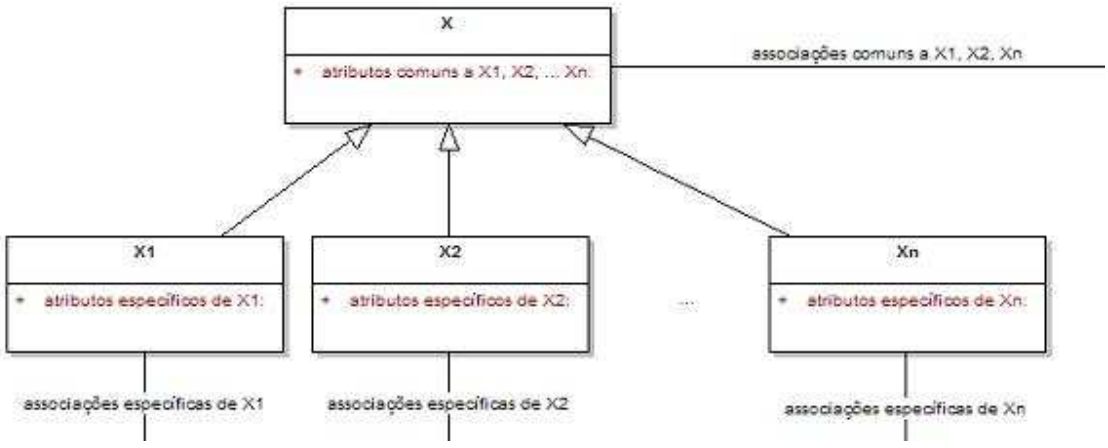


Figura 5.21: Representação esquemática de uma situação onde herança deveria ser usada.

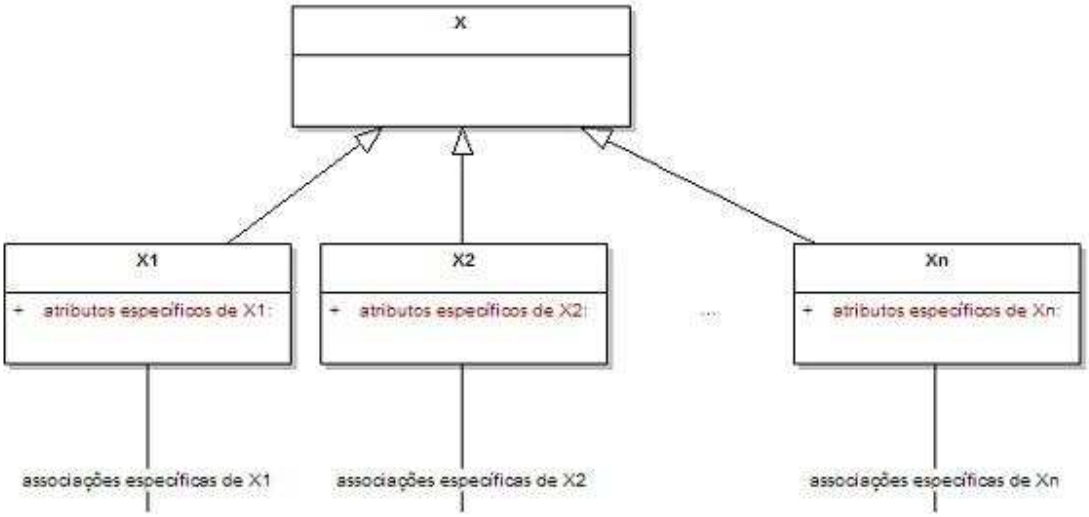


Figura 5.22: Representação esquemática de uma situação onde herança não deveria ser usada, pois não existem propriedades generalizadas.

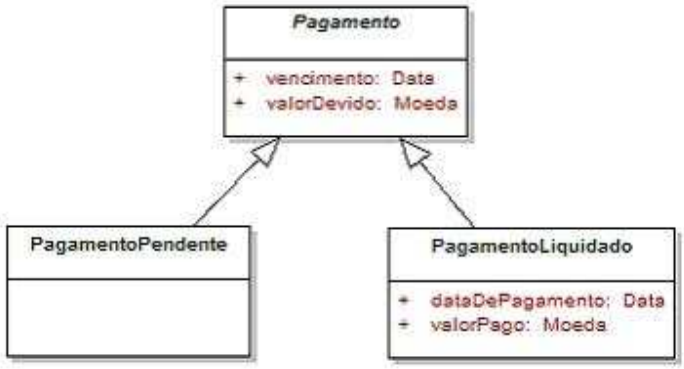


Figura 5.26: Forma inconveniente de modelar estados com herança.

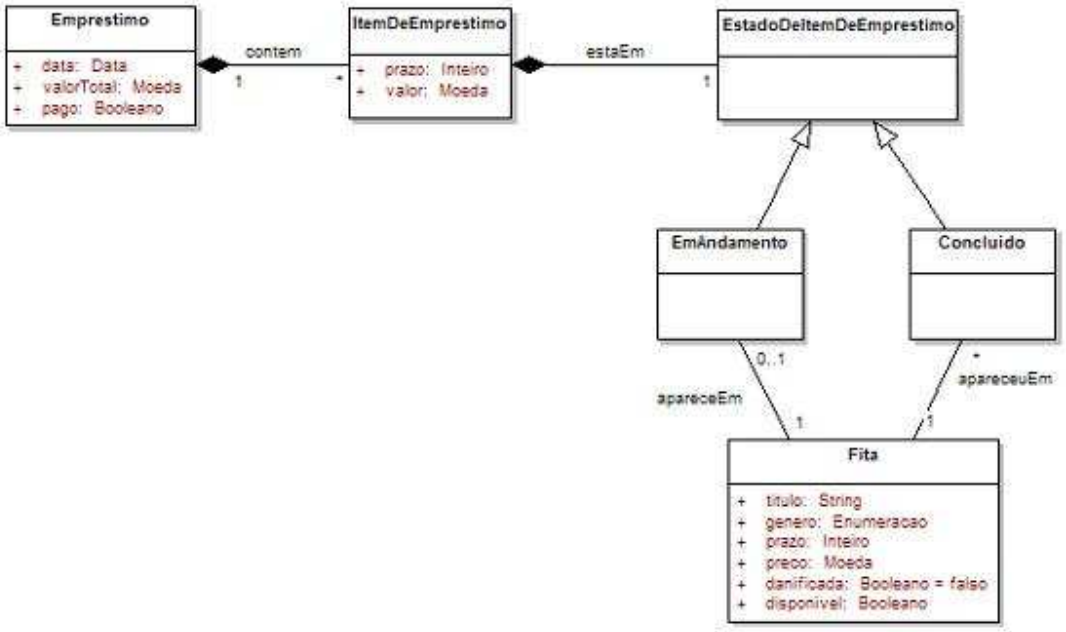


Figura 5.30: Modelagem de classes modais com transição não-monotônica usando o padrão de projeto Estado.

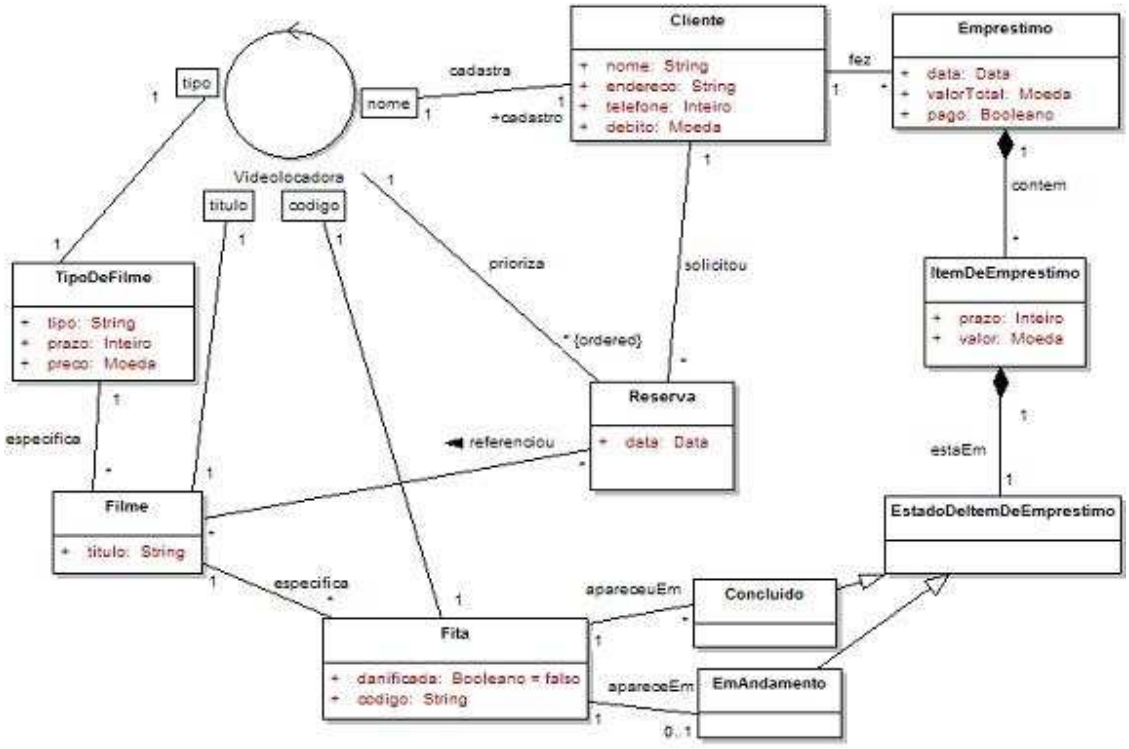
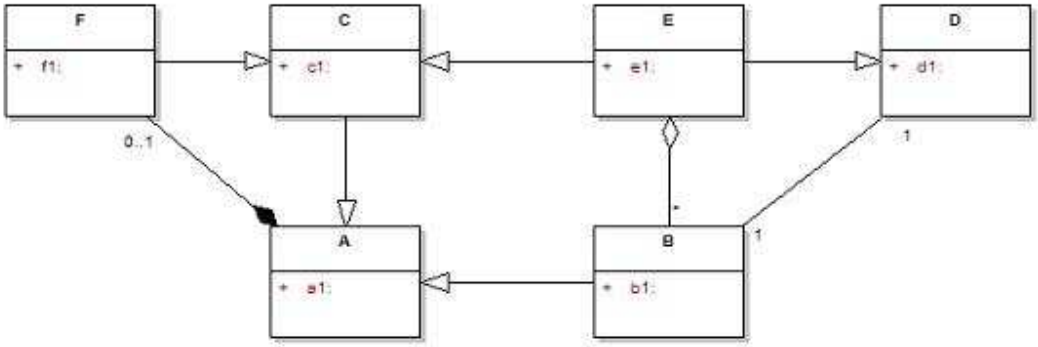


Figura 5.35: Um modelo conceitual para o primeiro ciclo da videolocadora.

Figura do exercício 3:



Capítulo 6 - Contratos.

(08/03/2005): Página 160, primeiro parágrafo. Redação correta: "Neste caso, o contrato da operação de sistema `identificaCliente` não teria nenhuma precondição, mas teria duas exceções **que deveriam ser** testadas durante a execução da operação.

(29/01/2007): Página 162, 3º parágrafo. Ao invés de "O modelo conceitual da Figura 5.31...", o correto é: "O modelo conceitual da Figura **5.35**...".

(08/03/2005): Na figura 6.24 a seta triangular de herança foi indevidamente substituída por uma seta simples que representa a associação direcional. Segue abaixo a figura correta.

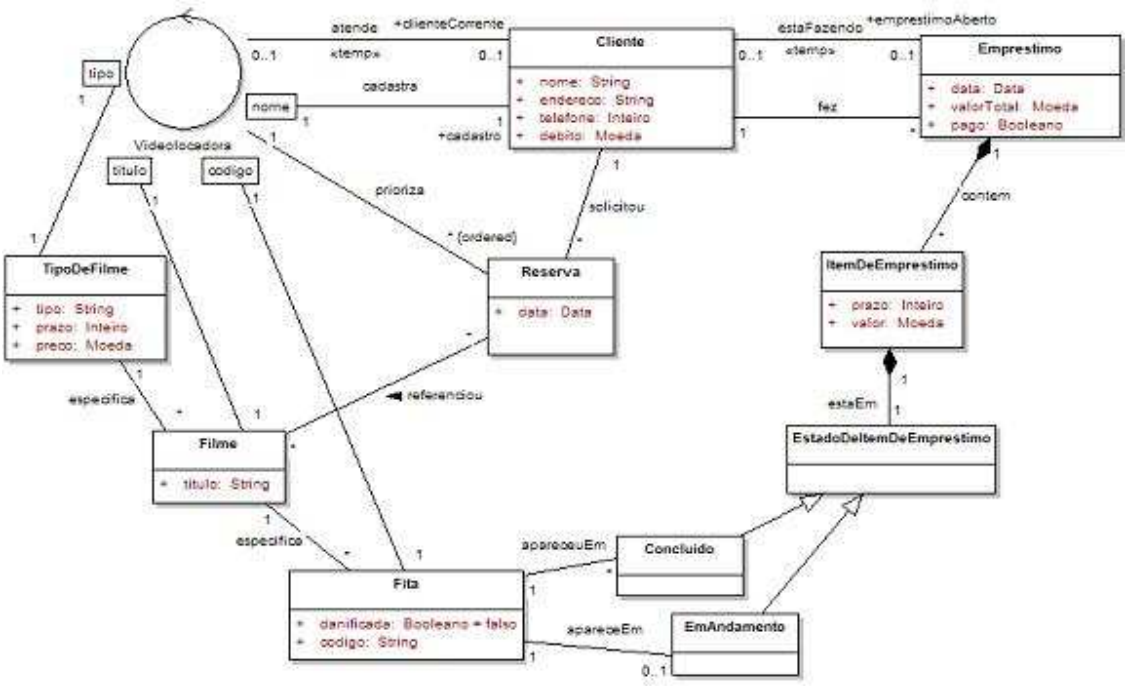


Figura 6.24: Modelo conceitual completo com associações temporárias.

Capítulo 7 - Projeto da Camada de Domínio.

(08/03/2005): Na figura 7.25 a seta triangular de herança foi indevidamente substituída por uma seta simples que representa a associação direcional. Segue abaixo a figura correta.

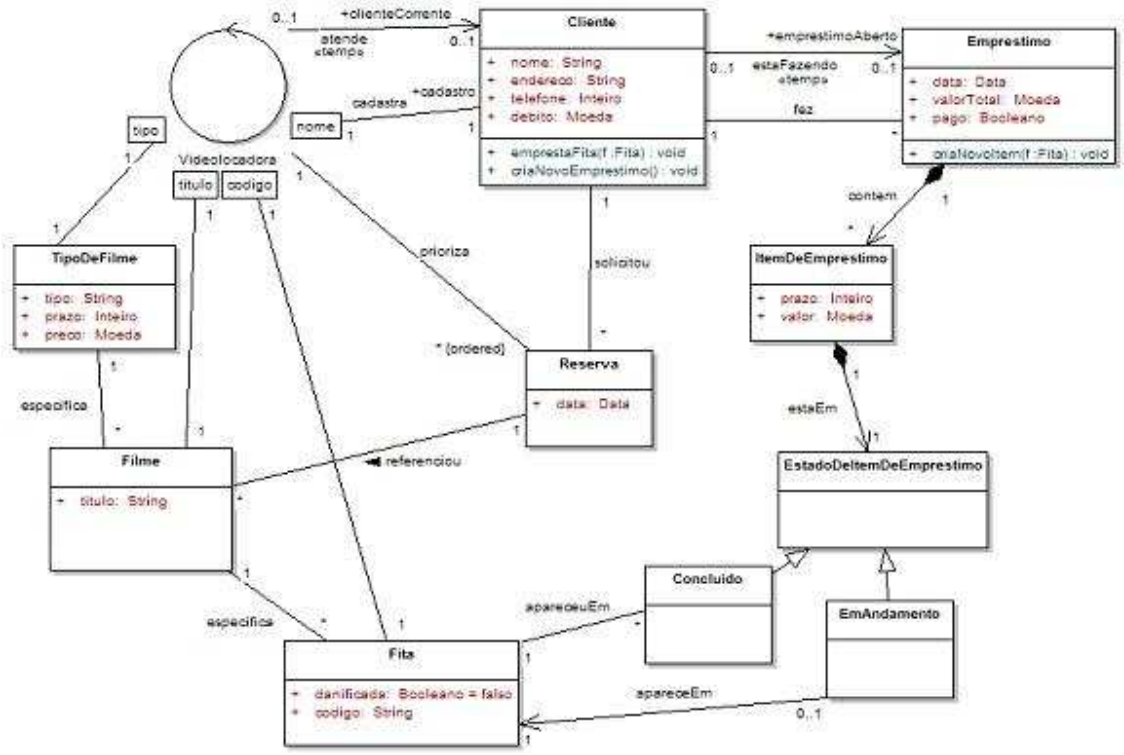


Figura 7.25: Diagrama de classes de projeto obtido a partir do modelo conceitual da figura 6.24 e do diagrama de colaboração da figura 7.23.

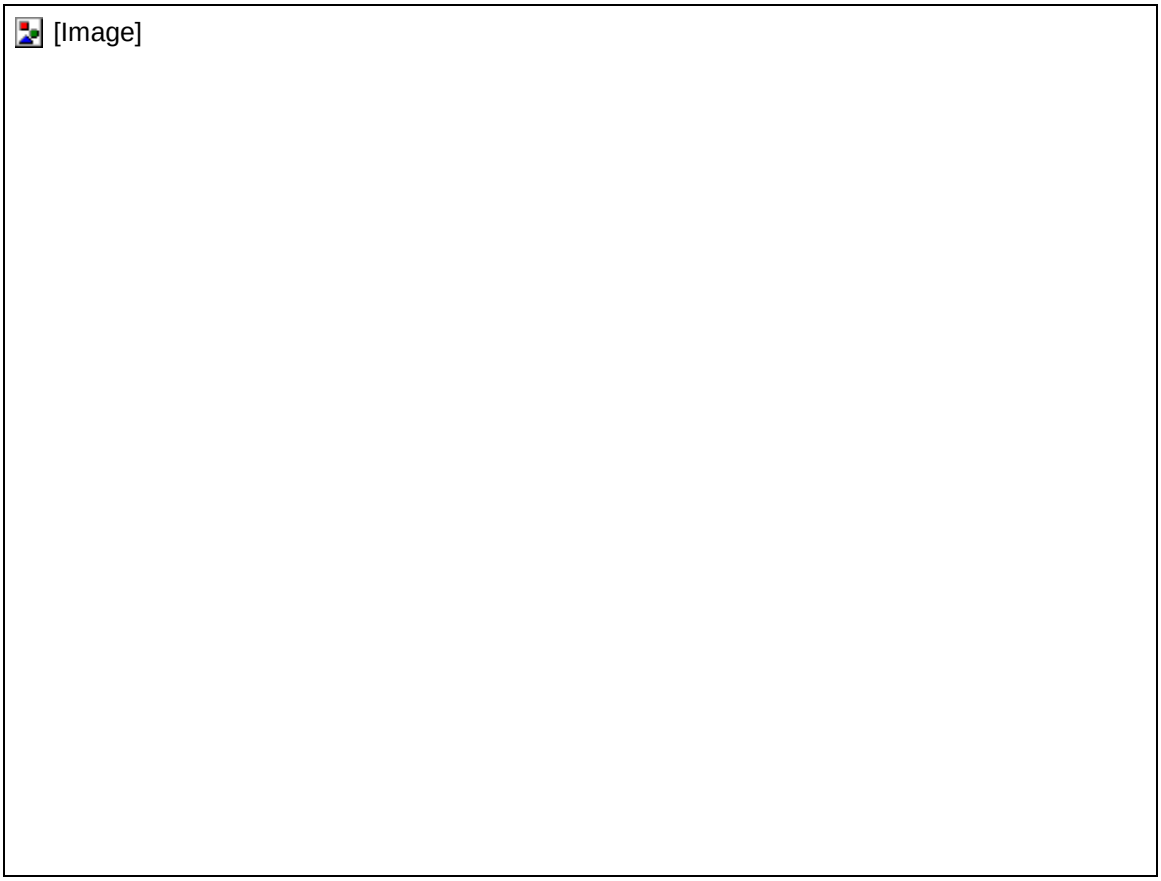
Capítulo 8 - Geração de Código.

(08/03/2005): Na figura 8.5 faltou um ";" no método "desassociaPagamento":


```
class Venda {
        private Pagamento pagamento;

        public Venda() { }
        public void associaPagamento(Pagamento pagamento) {
                this.pagamento = pagamento;
        }
        public void desassociaPagamento() {
                this.pagamento = null;
        }
        public Pagamento getPagamento() {
                return pagamento;
        }
}
```

(08/03/2005): O código na figura 8.6 poderia refletir melhor as características de Java (`HashSet`). O uso de `Collections.unmodifiableSet` impediria que o conjunto original fosse alterado externamente ao objeto `Cliente`.



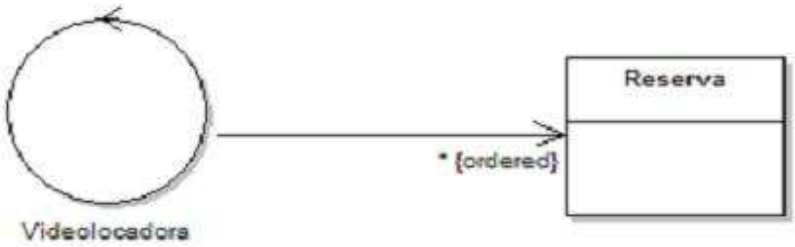
(08/03/2005): Página 238, último parágrafo. Redação correta: "Observe que neste caso, a variável que contém a associação tem seu nome definido no plural e seu tipo é `Set`. Assume-se que a classe `Set`, usada para representar o conjunto, possua um método **adiciona**, para incluir elementos no conjunto e um método `remove` para remover elementos do conjunto."

(08/03/2005): A figura 8.7 foi adequada para características específicas de Java, com o uso de equals, iterator, hashset e typecast.

```
public Set getEmprestimosNaData(Date data) {
        private Set empND = new HashSet();
        private Emprestimo atual;

        Iterator inter = emprestimos.iterator();
        while (inter.hasNext()) {
                atual = (Emprestimo) inter.next();
                if (atual.data().equals(data)) {
                        empND.add(atual);
                };
        };
        return empND;
}
```

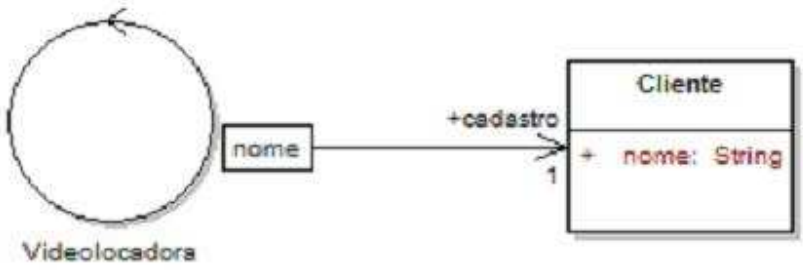
(08/03/2005): Na figura 8.8, uma série de pequenos erros no código foram corrigidos.




```
class Videolocadora {
        private List reservas = new ArrayList();

        public Videolocadora() { }
        public void adicionaReserva(Reserva reserva, int posicao) {
                this.reservas.add(posicao,reserva);
        }
        public void removeReserva(Reserva reserva) {
                this.reservas.remove(reserva);
        }
        public void removeReservaNaPosicao(int posicao) {
                this.reservas.remove(posicao);
        }
        public List getReservas() {
                return Collections.unmodifiableList(reservas);
        }
        public Reserva getReservaNaPosicao(int posicao) {
                return (Reserva) reservas.get(posicao);
        }
}
```

(08/03/2005): Uma série de pequenos erros foram corrigidos na figura 8.9:




```
class Videolocadora {
        private Map cadastro = new HashMap();

        public Videolocadora () { }
        public void adicionaNoCadastro(Cliente cliente) {
                this.cadastro.put(cliente.getNome(), cliente);
        }
        public void removeDoCadastro(Cliente cliente) {
                this.removeDoCadastroPorNome(cliente.getNome());
        }
        public void removeDoCadastroPorNome(String nome) {
                this.cadastro.remove(nome);
        }
        public Collection getCadastro(){
                return cadastro.values();
        }
        public Cliente getCliente(String nome) {
                return (Cliente) cadastro.get(nome);
        }
}
```

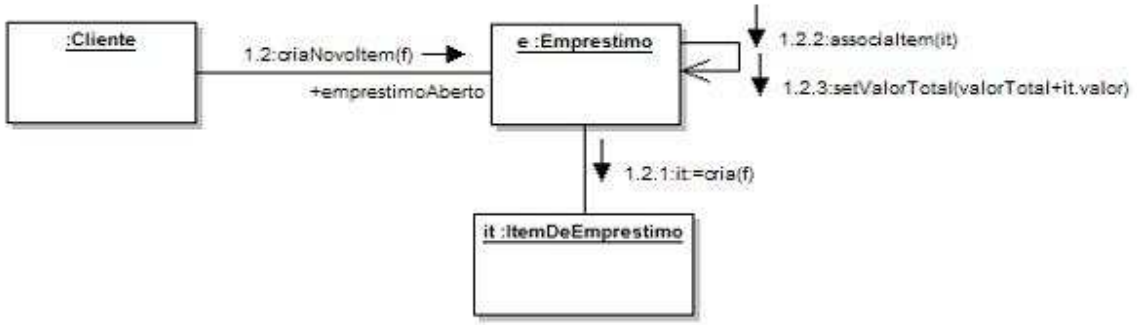
(08/03/2005): Página 242, último parágrafo. Redação correta: "Veja que o mapeamento é feito de String (atributo nome) para Cliente, sendo o conjunto de nomes **acessado** por cadastro.**keySet**() e o conjunto de clientes por cadastro.**values**(). É possível implementar pelo menos duas operações de remoção: pela chave e pelo valor. Além disso, é possível implementar uma consulta para obter um cliente específico dado um nome de cliente (getCliente).

(08/03/2005): Na figura 8.11, o emprestimo era explicitamente destruído, o que não é necessário fazer em Java:


```
class Cliente {
        private Set emprestimos = new HashSet();

        public Cliente () {}
        public void adicionaEmprestimoAux(Emprestimo emprestimo) {
                emprestimos.add(emprestimo);
        }
        public void removeEmprestimoAux(Emprestimo emprestimo) {
                emprestimos.remove(emprestimo);
        }
        public void adicionaEmprestimo(Emprestimo emprestimo) {
                if (emprestimo.getCliente() != null) {
                        emprestimo.getCliente().removeEmprestimoAux(emprestimo);
                };
                this.adicionaEmprestimoAux(emprestimo);
                emprestimo.associaClienteAux(this);
        }
        public void removeEmprestimo(Emprestimo emprestimo) {
                this.removeEmprestimoAux(emprestimo);
        }
        public Set getEmprestimos() {
                return emprestimos;
        }
}
```

(08/03/2005): Na figura 8.15 foi adicionada a inicialização da coleção e removido um "private" desnecessário.




```
class Emprestimo {
        private Set itensDeEmprestimo = new HashSet();

        public void criaNovoItem(Fita f) {
                ItemDeEmprestimo it = new ItemDeEmprestimo(f);

                this.associaItem(it);
                this.setValorTotal(this.getValorTotal()+it.getValor());
        }
}
```

Capítulo 9 - Projeto da Camada de Interface.

Capítulo 10 - Camada de Persistência.

(08/03/2005): Na figura 10.7 a seta triangular de herança foi indevidamente substituída por uma seta simples que representa a associação direcional. Segue abaixo a figura correta.

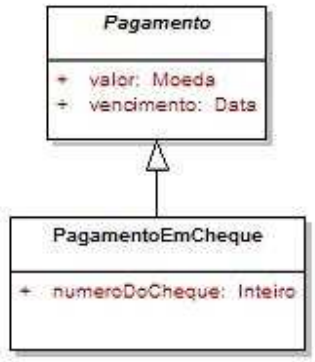


Tabela: PagamentoEmCheque			
#IUOPagamentoEmCheque (chave)	**numeroDoCheque**	**valor**	**vencimento**
7887879	8768769	60,00	13/02/04
9876978	6743680	12,00	15/02/04
9877655	8976876	99,88	12/05/03
8976678	4354545	80,90	17/09/04

Figura 10.7: Tabela representando classe com herança.

FIM.