

Documento de Projeto de Software

Projeto: Vídeo Locadora Passatempo

Versão: 1.0

Responsável: Ricardo de Almeida Falbo

1. Introdução

Este documento apresenta o documento de projeto (design) do sistema de apoio às atividades da Vídeo Locadora Passatempo. Essa atividade foi conduzida em refinamentos sucessivos, começando pelo projeto da arquitetura do sistema, passando ao detalhamento dos componentes da arquitetura, até chegar ao projeto detalhado das classes. Este documento está organizado da seguinte forma: a seção 2 apresenta a plataforma de software a ser utilizada na implementação do sistema; a seção 3 discute aspectos do projeto da arquitetura do sistema; as seções 4 e 5 apresentam os modelos relativos aos subsistemas identificados; finalmente, a seção 6 discute aspectos do projeto de classes.

2. Plataforma de Implementação

O sistema em questão trata-se de um Sistema de Informação e apresenta as seguintes características:

- Envolve grande quantidade de dados e a sua gerência deve ser feita usando um banco de dados;
- Usuários acessam os dados concorrentemente. Há funcionalidades que estarão disponíveis pela Internet e haverá pelo menos dois postos de trabalho dentro da locadora para atendimento a clientes;
- Há uma grande quantidade de interfaces com o usuário;
- O sistema precisa estar integrado com o sistema da administradora de cartão de crédito.

Levando-se em consideração essas características, decidiu-se implementar o sistema para a videolocadora Passatempo usando a linguagem de programação Java, o banco de dados relacional PostgreSQL e o *framework* de mapeamento objeto-relacional Hibernate.

3. Arquitetura de Software

Como se pode perceber pela especificação de requisitos para o sistema em questão, não há grandes restrições de desempenho e disponibilidade, ainda que algumas restrições tenham sido explicitamente apontadas. Assim, levando-se em consideração os requisitos para o sistema proposto, foram considerados como os principais atributos de qualidade a

serem incorporados ao sistema os seguintes, apresentados juntamente com as táticas a serem aplicadas:

- Usabilidade:
 - Separar a interface do restante da aplicação.
 - Prover ao usuário a capacidade de entrar com comandos que permitam operar o sistema de modo mais eficiente. Para tal, as interfaces do sistema devem permitir, sempre que possível, a entrada por meio de seleção ao invés da digitação de campos.
- Manutenibilidade
 - Coerência semântica: a organização do sistema deve se dar de modo que as responsabilidades em um módulo trabalhem em conjunto sem depender excessivamente de outros módulos;
 - Uso de interfaces com ocultação de informações específicas sobre a implementação dos módulos;
 - Uso de um intermediário para isolar o mecanismo de persistência de dados;
 - Uso de um intermediário para tratar as requisições da interface.
- Segurança:
 - Autenticar usuários usando *login* e senha;
 - Autorizar usuários, criando os seguintes grupos: (i) Gerente de Acervo – acesso às funcionalidades do controle de acervo; (ii) Atendente – acesso às funcionalidades de atendimento a clientes; (iii) Administrador – acesso geral a todas as funcionalidades do sistema, incluindo o cadastro de usuários.
 - Limitar a exposição, disponibilizando pela Internet somente funcionalidades de consulta ao acervo.
 - Manter uma trilha de auditoria para as operações de atendimento a cliente, sempre registrando o atendente que efetuou uma locação ou devolução (e, por conseguinte, um pagamento).

Ainda que os demais atributos de qualidade não tenham sido considerados como sendo condutores da arquitetura, algumas táticas foram aplicadas visando garantir o nível de atendimento requerido. A seguir, as táticas consideradas são listadas:

- Desempenho:
 - Reduzir overhead computacional em situações que não comprometam a manutenibilidade.
 - Estabelecer uma configuração de hardware mínima para comportar o sistema.
- Disponibilidade: uso de exceções e transações para detecção, tratamento e prevenção de falhas.
- Portabilidade: uso da linguagem Java e de bibliotecas e mecanismos de persistência capazes de rodar nos sistemas operacionais Windows e Linux.

Tomando por base as características do sistema discutidas na seção 2 e os atributos de qualidade e táticas selecionadas para tratá-los apresentados anteriormente, decidiu-se adotar um estilo combinando camadas e partições.

Inicialmente, duas partições principais foram definidas, procurando-se preservar a divisão em subsistemas realizada na fase de análise. Cada uma dessas partições, por sua vez, está organizada em três camadas, a saber: camadas de Interface com o Usuário (ciu), Lógica de Negócio (cln) e Gerência de Dados (cgd), como mostra a Figura 1.

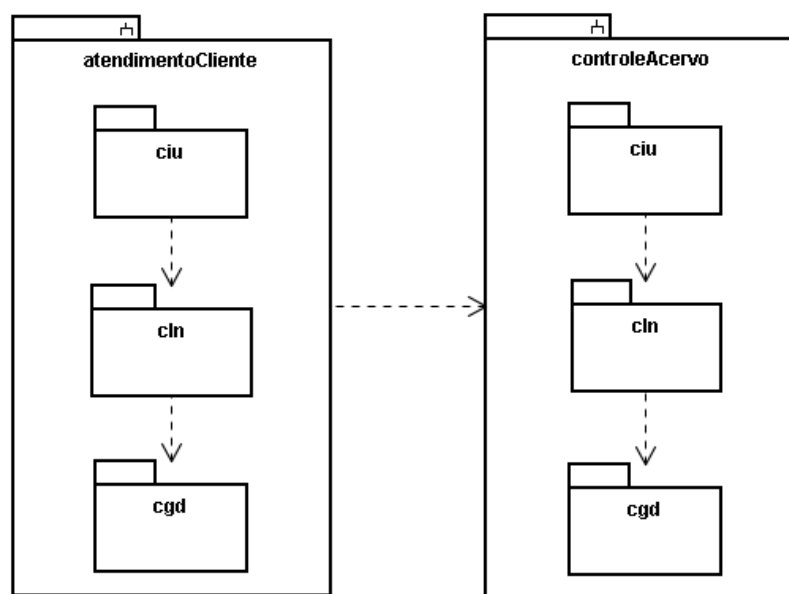


Figura 1 – Arquitetura de Software Inicial.

Com o detalhamento do projeto, em função de algumas decisões, a arquitetura originalmente proposta sofreu algumas alterações, a saber:

- No projeto do CLN, optou-se por usar o padrão Camada de Serviço. Assim, o CLN foi subdividido em dois pacotes: Componente de Domínio do Problema (cdp) e Componente de Gerência de Tarefas (cgt).
- Uma vez que há a necessidade de ter parte do sistema rodando na Web, essa porção foi deslocada para uma nova partição, o subsistema Consulta ao Acervo (consultaAcervo).
- Buscando o desenvolvimento para e com reúso, foram reutilizados os Utilitários Pessoa (utilitarioPessoa) e Persistência (utilitarioPersistencia) e foi criado o *framework* Pagamento (utilitarioPagamento)¹.
- Por fim, as funcionalidades básicas gerais de interface do sistema Videolocadora (controlador do sistema e janela principal) foram separadas no pacote videolocadora.

A Figura 2 mostra o projeto completo da arquitetura de software do sistema Videolocadora. A seguir, o projeto de cada uma dessas partições é apresentado.

¹ Neste documento foi omitido o Utilitário Segurança

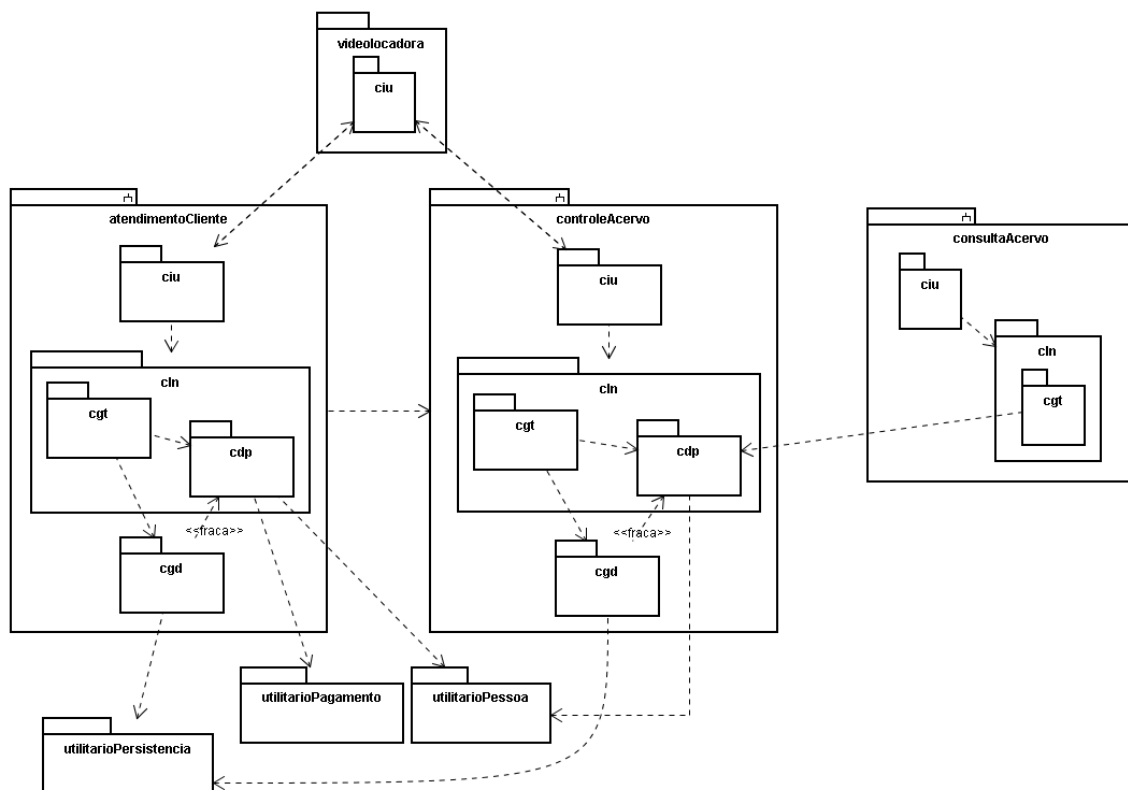


Figura 2 – Arquitetura de Software Completa.

Vale ressaltar que a dependência entre os pacotes CGD e CDP existe apenas para instanciar objetos recuperados do banco de dados. Nenhum outro serviço é utilizado e, portanto, esta é uma dependência fraca.

4. Subsistema Controle de Acervo

Conforme discutido anteriormente, o subsistema Controle de Acervo está organizado em três camadas: Camada de Lógica de Negócio, Camada de Interface com o Usuário e Camada de Gerência de Dados.

4.1 – Camada de Lógica de Negócio

Para organizar a camada de lógica de negócio deste pacote, foi escolhido o padrão Camada de Serviço. Sendo assim, essa camada é dividida em dois componentes: Componente de Domínio do Problema (cdp) e Componente de Gerência de Tarefas (cgt), como mostra a Figura 2. Esse padrão utiliza um intermediário para tratar as requisições da interface (o cgt), conforme tática definida para tratar a manutenibilidade. A seguir, o projeto do Componente de Domínio do Problema (cdp) é apresentado.

4.1.1 – Componente de Domínio do Problema (CDP)

A Figura 3 apresenta o diagrama de classes do CDP do subsistema Controle de Acervo. Uma vez que essencialmente as funcionalidades providas por esse componente são de cadastros básicos, não foram elaborados diagramas de sequência.

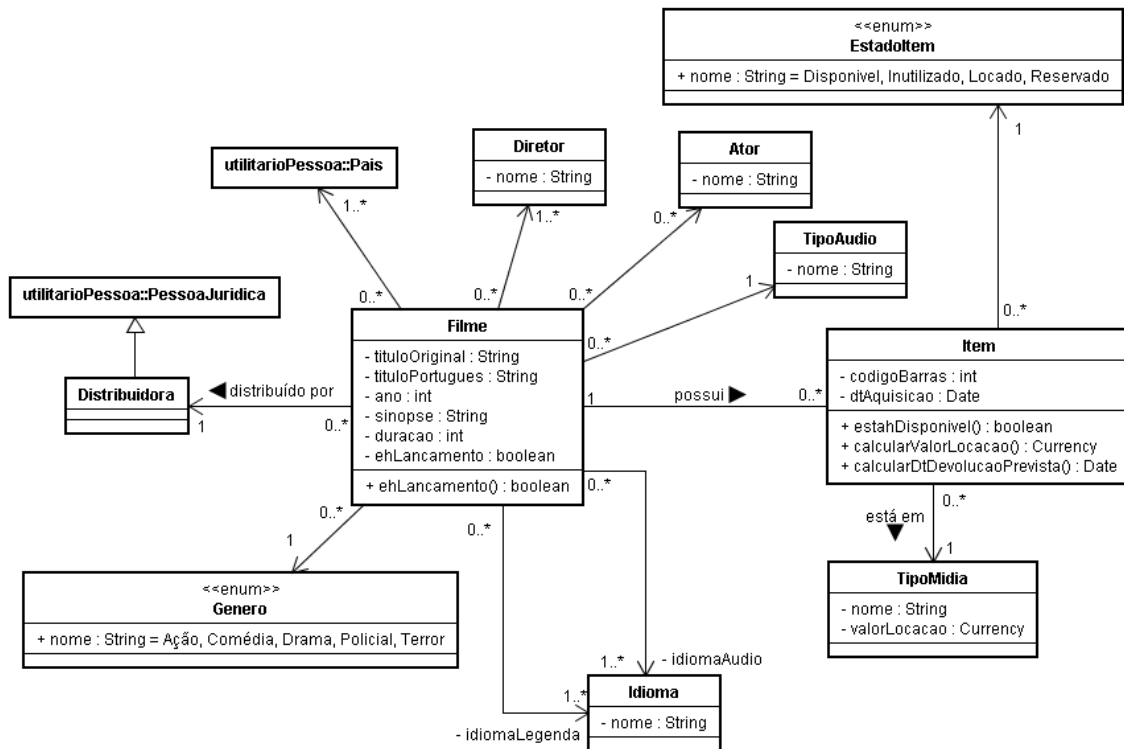


Figura 3 – Diagrama de Classes do CDP do Subsistema Controle de Acervo

As classes **Diretor**, **Ator**, **TipoAudio** e **Idioma**, bem como os tipos enumerados **Genero** e **EstadoItem** foram introduzidos visando à usabilidade (tática “Prover ao usuário a capacidade de entrar com comandos que permitam operar o sistema de modo mais eficiente”). Procurou-se, ainda, reutilizar classes previamente projetadas do Utilitário Pessoa.

4.1.2 – Componente de Gerência de Tarefas (CGT)

No projeto do CGT, optou-se por criar uma classe gerenciadora de tarefa para cada caso de uso identificado na fase de análise. A classe **ApICadastrarFilme** trata do caso de uso Cadastrar Filme e dos casos de uso fortemente relacionados a ele identificados na fase de projeto, a saber: Cadastrar Ator, Cadastrar Diretor, Cadastrar Tipo de Áudio, Cadastrar País e Cadastrar Idioma. Esses casos de uso foram agrupados em uma única classe de aplicação, devido ao fato de, ao se cadastrar um filme, poder se querer cadastrar novas informações de atores, diretores, tipos de áudio, países e idiomas. As demais classes lidam com seus respectivos casos de uso.

Uma vez que o projeto do CGT está fortemente relacionado ao projeto da Interface com o Usuário, um único diagrama foi elaborado, o qual é mostrado na Figura 4.

4.2 – Camada de Interface com o Usuário

Para organizar a camada de interface com o usuário, foi adotado o padrão Modelo-Visão-Controlador. Sendo assim, essa camada possui classes de visão, mostradas no diagrama da Figura 4 com o estereótipo de classes de fronteira (<<boundary>>) da UML e destacadas em amarelo, e classes de controle de interação, mostradas no diagrama da Figura 4 com o estereótipo de classes controladoras (<<control>>) da UML e destacadas em vermelho.

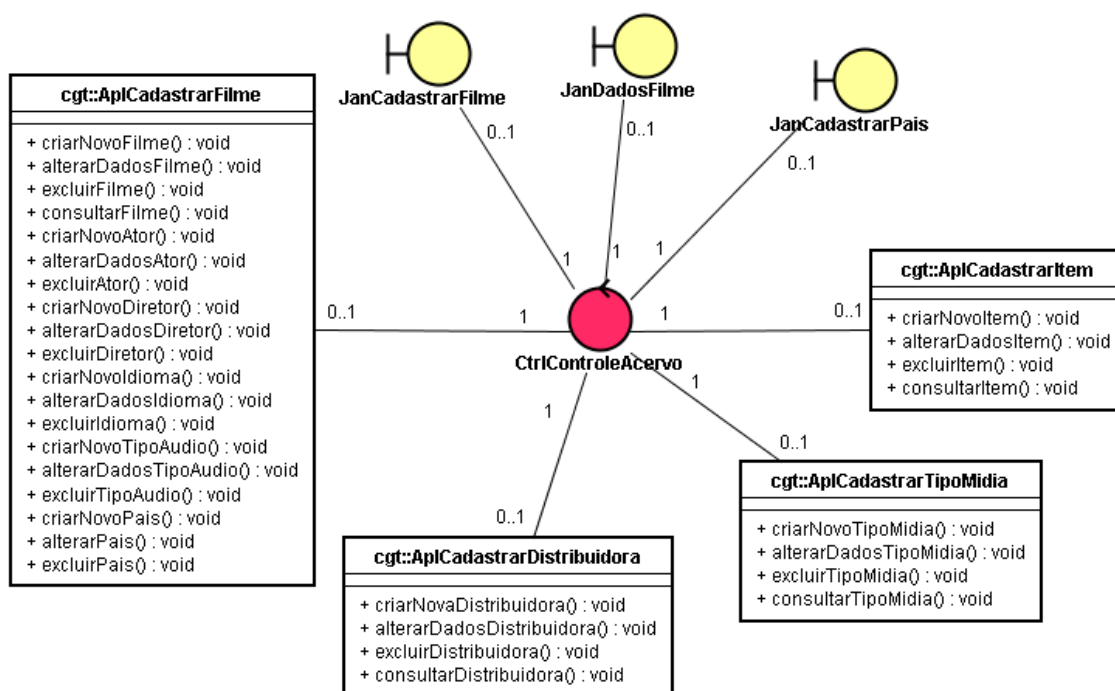


Figura 4 – Diagrama de Classes (parcial) do CIU do Subsistema Controle de Acervo

Decidiu-se utilizar uma única classe controladora de interação para controlar todo esse subsistema, uma vez que a classe controladora é bastante simples, fazendo a ligação entre as classes de visão e as classes gerenciadoras de tarefa (modelo no padrão MVC).

Vale ressaltar que o diagrama da Figura 4 é apenas parcial, tendo em vista que apenas algumas classes de visão foram apresentadas. Para o projeto das classes de visão, optou-se por elaborar protótipos e mostrar apenas os layouts das classes mostradas na Figura 4. Assim, as figuras 5, 6 e 7 mostram, respectivamente, os layouts das classes JanCadastrarFilme, JanDadosFilme e JanCadastrarPais².

² Observar que essas janelas não correspondem efetivamente ao projeto das janelas necessárias para o sistema da Videolocadora Passatempo. Elas são meramente ilustrativas de como seriam janelas deste tipo. Os layouts mostrados foram, na verdade, extraídos do sistema VideoLoc 1.4, disponível em <http://www.baixaki.com.br/download/videoloc.htm>.

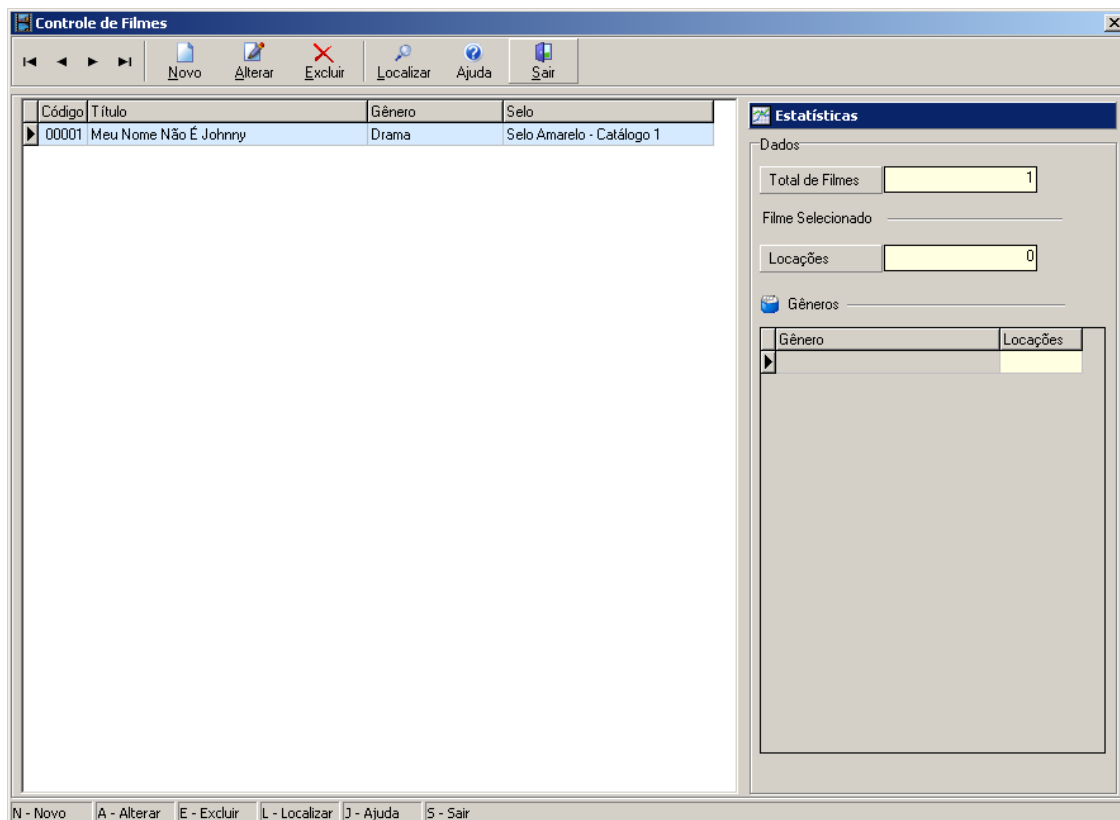


Figura 5 – Layout ilustrativo da janela JanCadastrarCliente.

The screenshot shows a window titled "Cadastro de filmes". It contains the following fields and controls:

- Código:** 00001
- Título:** Meu Nome Não É Johnny
- Ano:** 2008
- Tradução:** (dropdown menu)
- Duração (minutos):** 126
- Mídia:** DVD
- Idioma Original:** Português
- País:** Brasil
- Gênero:** Drama
- Selo:** (dropdown menu)
- Elenco:** Cássia Kiss, Cleo Pires, Selton Mello, Júlia Lemmertz, Eva Todor, Giulio Lopes
- Sinopse:** Mario Lima volta com um filme que trata de um tema polêmico e forte ao contar a história de João Guilherme Estrella, um traficante de classe média alta na sociedade carioca dos anos 1980. O filme é baseado no livro de Guilherme Fiúza e o ator Selton Mello (de O Cheiro do Ralo) foi o escolhido
- Buttons:** Gravar, Novo, Replicar, Sair

Figura 6 – Layout ilustrativo da janela JanDadosCliente.

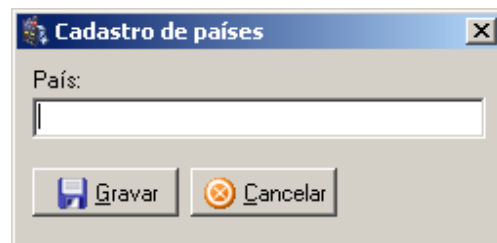


Figura 7 – Layout ilustrativo da janela JanCadastrarPaís.

4.3 – Camada de Gerência de Dados

A persistência dos objetos deste sistema é realizada em um banco de dados relacional, utilizando o *framework* de persistência Hibernate, sendo desejável isolar os impactos da tecnologia de bancos de dados sobre o sistema. Assim, optou-se por adotar o Padrão DAO e foi utilizado o utilitário de Persistência, apresentado na Figura 8, que já trata vários dos aspectos desse padrão.

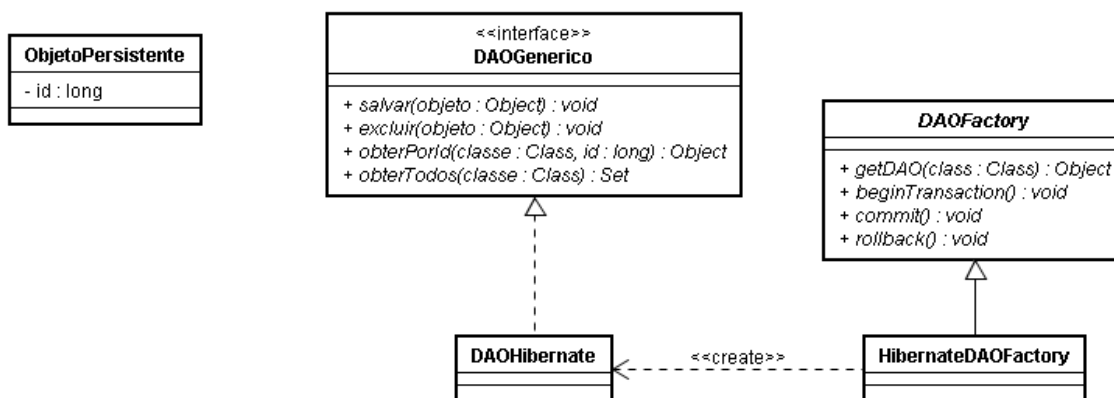


Figura 8 – Infraestrutura Genérica de Persistência.

Classes a serem persistidas devem herdar da classe `ObjetoPersistente`, que provê identificadores únicos para os objetos (Ids) a serem usados para mapear objetos em memória com as correspondentes linhas das tabelas. Para cada classe de domínio a ser persistida, devem ser criadas uma classe DAO e uma interface DAO correspondente. A primeira deve herdar de `DAOHibernate`, uma classe genérica que possui as funcionalidades básicas de acesso ao mecanismo de persistência, e deve implementar a interface DAO associada, provendo flexibilidade no momento da criação de novas operações especializadas para um certo elemento de domínio. Já a interface DAO da classe a ser persistida deve herdar da interface genérica `DAOGenerico`. Seguindo essa abordagem, cada classe a ser persistida tem uma correspondente classe controladora da lógica de persistência (ou classe mapeadora), responsável pela interação com o banco de dados relacional, e implementa uma interface correspondente, como mostra a Figura 9.

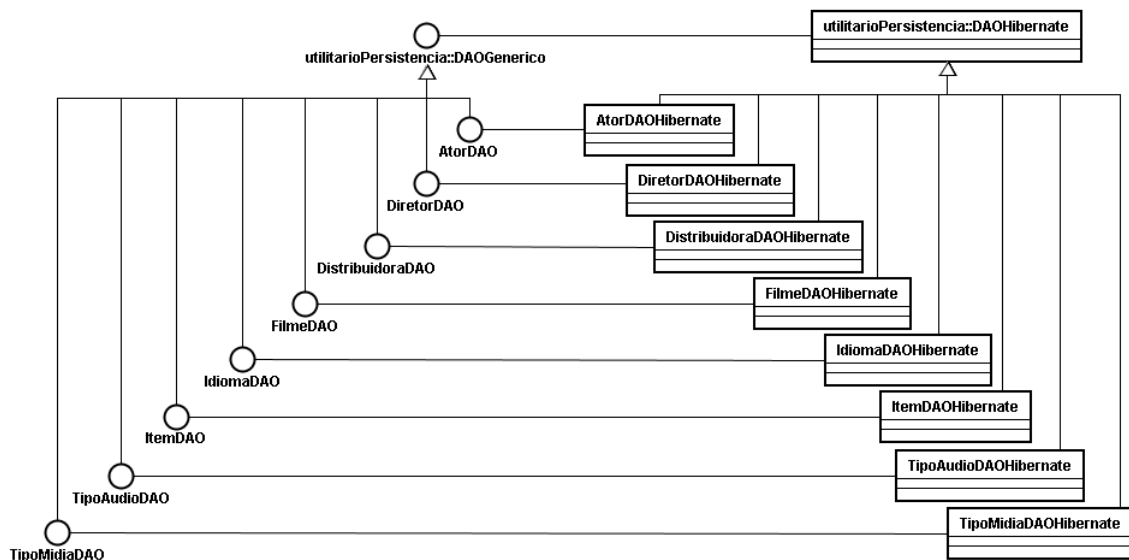


Figura 9 – CGD do pacote Controle de Acervo.

5. Subsistema Atendimento a Cliente

Conforme discutido anteriormente, assim como o subsistema Controle de Acervo, o subsistema Atendimento a Cliente está organizado em três camadas: Camada de Lógica de Negócio, Camada de Interface com o Usuário e Camada de Gerência de Dados.

5.1 – Camada de Lógica de Negócio

Para organizar a camada de lógica de negócio deste pacote, foi escolhido o padrão Camada de Serviço. Sendo assim, essa camada é dividida em dois componentes: Componente de Domínio do Problema (cdp) e Componente de Gerência de Tarefas (cgt), como mostra a Figura 2. A seguir, o projeto de cada um desses componentes é discutido.

5.1.1 – Componente de Domínio do Problema (CDP)

A Figura 10 apresenta o diagrama de classes do CDP do subsistema Atendimento a Cliente. As operações mostradas nesse diagrama e no diagrama de classes do CDP do subsistema Controle de Acervo foram definidas a partir da elaboração do diagrama de sequência mostrado na Figura 11.

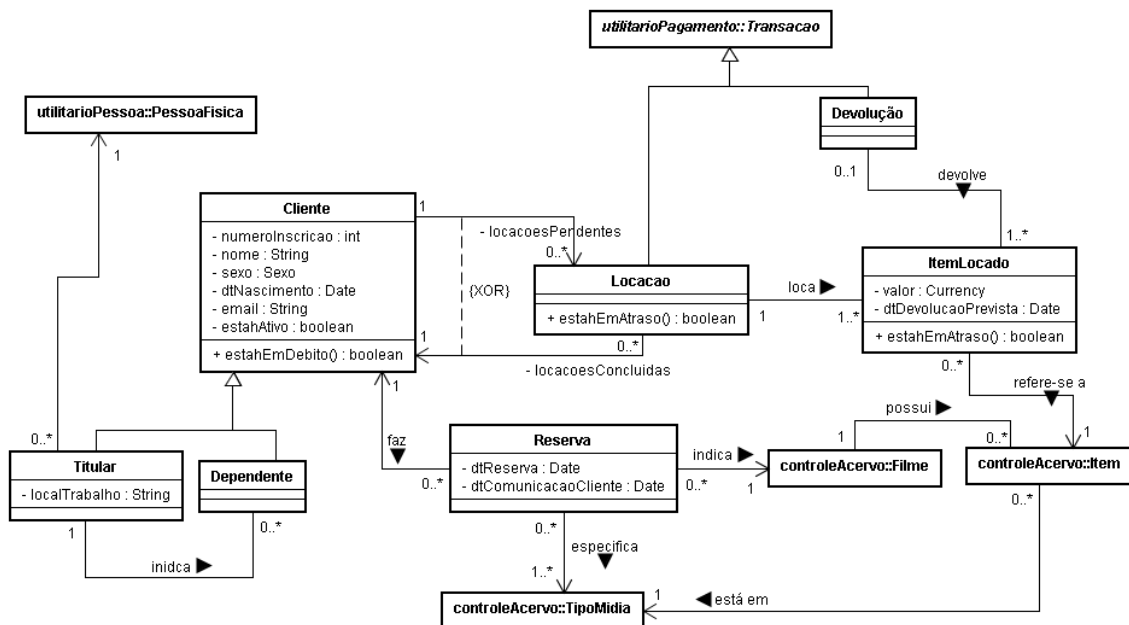


Figura 10 – Diagrama de Classes do CDP do Subsistema Atendimento a Cliente

Vale a pena registrar os fatores que motivaram algumas das decisões tomadas no projeto deste componente. Primeiro, com vistas ao desenvolvimento para reuso, foi criado um Utilitário Pagamento, capturando aspectos gerais de pagamento de transações em geral, como mostra a Figura 12. Segundo, para reutilizar a classe PessoaFisica do utilitário Pessoa (ver Figura 13), foi usada uma abordagem de delegação por associação. Por fim, de modo a não comprometer o desempenho do sistema, a associação entre Cliente e Locacao foi desmembrada em duas associações, uma tratando apenas das locações pendentes e a outra tratando das locações já concluídas.

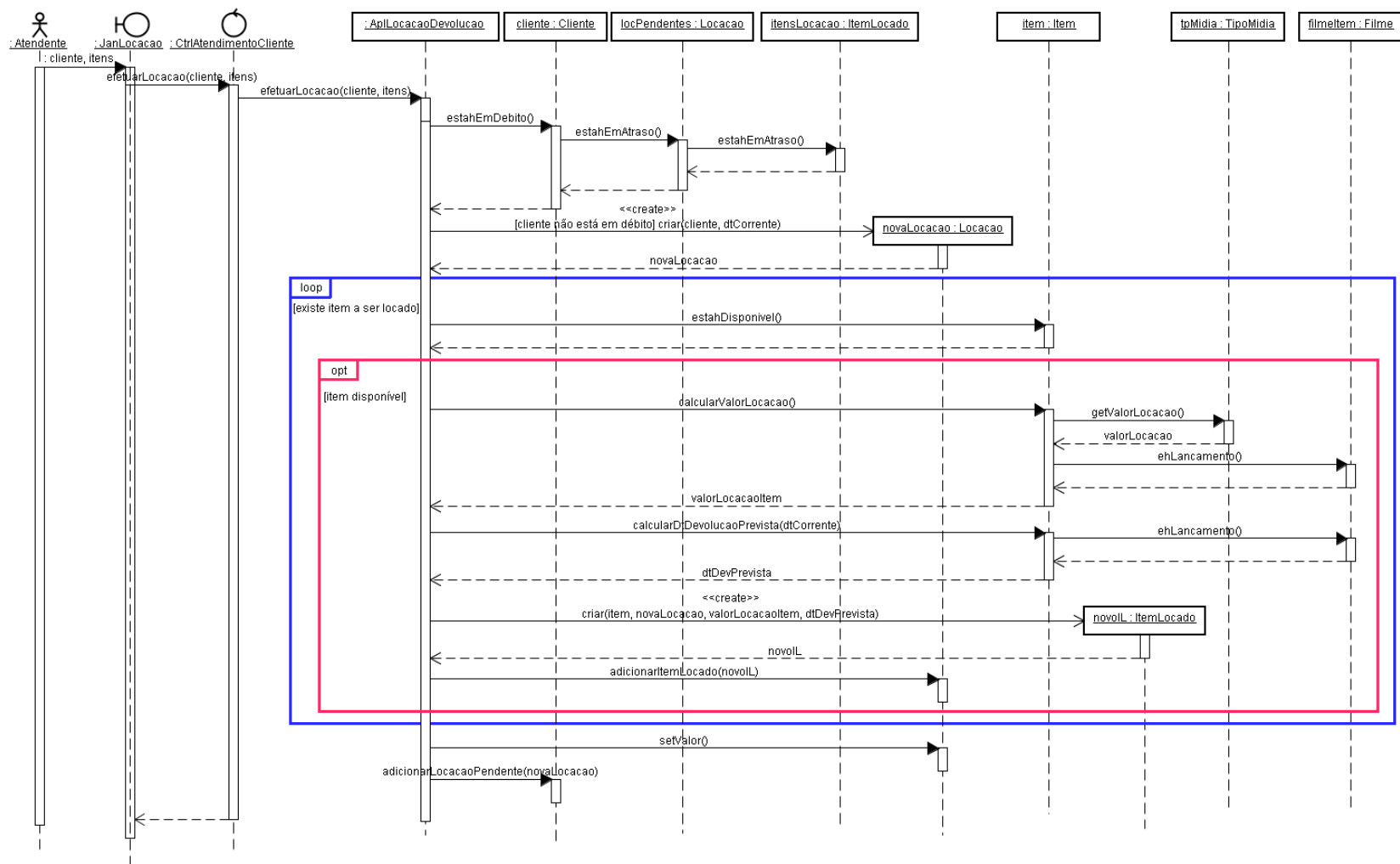


Figura 11 – Diagrama de Sequência para o fluxo de eventos Efetuar Nova Locação do caso de uso Efetuar Locação.

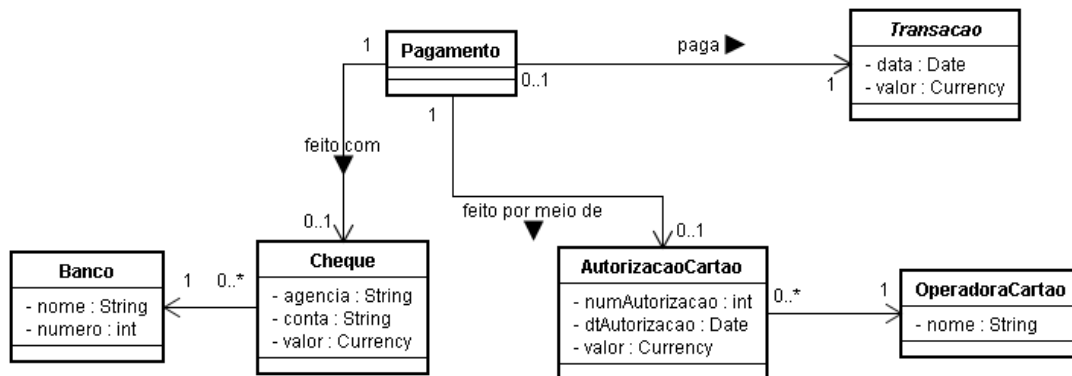


Figura 12 – Utilitário Pagamento

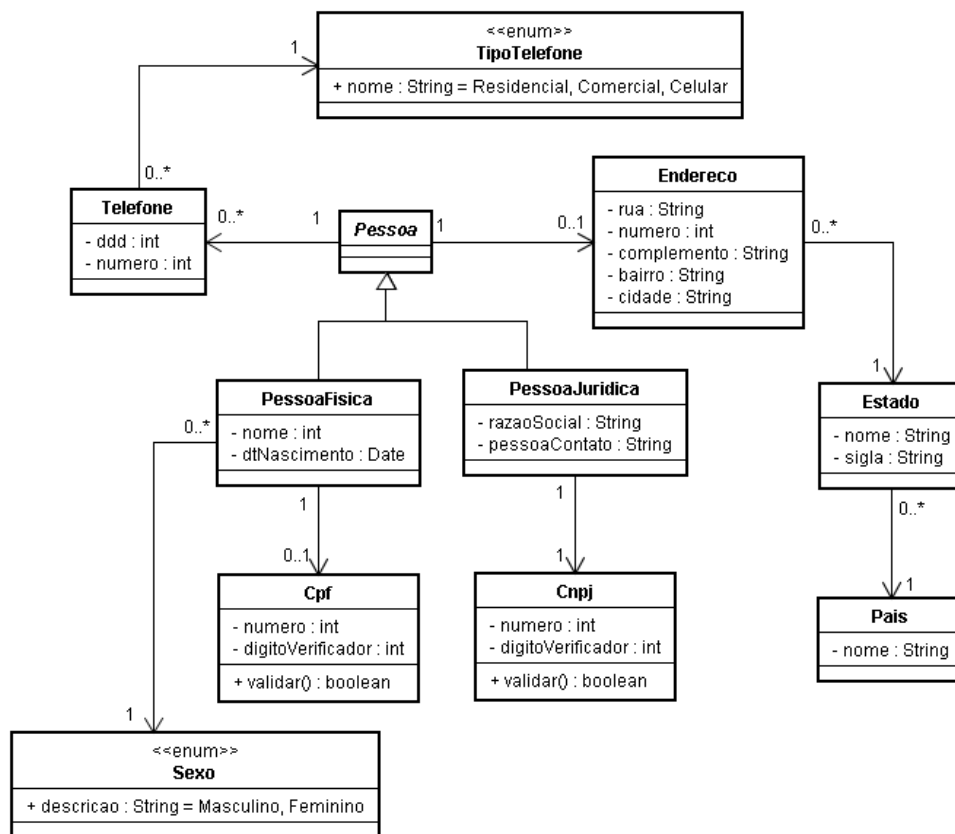


Figura 13 – Utilitário Pessoa

5.1.2 – Componente de Gerência de Tarefas (CGT)

No projeto do CGT, optou-se por criar uma classe gerenciadora de tarefa para cada caso de uso identificado na fase de análise. A exceção fica por conta da classe `AplLocacaoDevolucao` que trata dos casos de uso Efetuar Locação e Efetuar Devolução. Essa decisão foi tomada, considerando que locação e devolução são funcionalidades bastante relacionadas. As demais classes lidam com seus respectivos casos de uso.

Uma vez que o projeto do CGT está fortemente relacionado ao projeto da Interface com o Usuário, um único diagrama foi elaborado, o qual é mostrado na Figura 14.

5.2 – Camada de Interface com o Usuário

Para organizar a camada de interface com o usuário, foi adotado o padrão Modelo-Visão-Controlador. Sendo assim, essa camada possui classes de visão, mostradas no diagrama da Figura 14 com o estereótipo de classes de fronteira (<<boundary>>) da UML e destacadas em amarelo, e classes de controle de interação, mostradas no diagrama da Figura 14 com o estereótipo de classes controladoras (<<control>>) da UML e destacadas em vermelho.

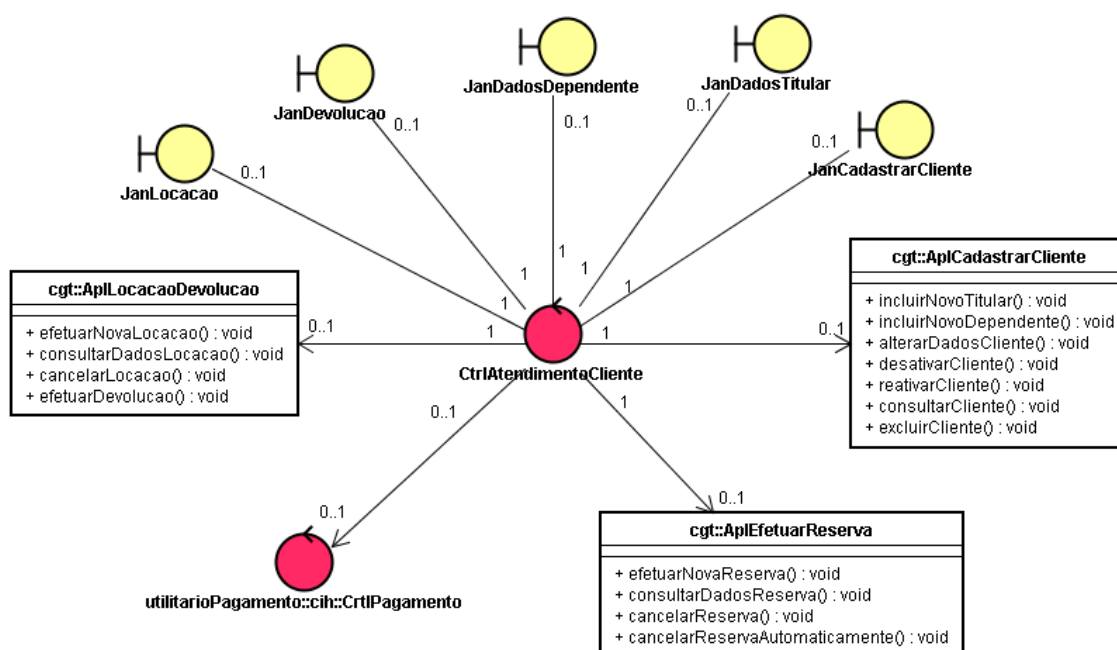


Figura 14 – Diagrama de Classes (parcial) do CIU do Subsistema Controle de Acervo

Decidiu-se utilizar uma única classe controladora de interação para controlar todo esse subsistema, uma vez que a classe controladora é bastante simples, fazendo a ligação entre as classes de visão e as classes gerenciadoras de tarefa (modelo no padrão MVC). A exceção fica por conta da criação da classe controladora de interação CtrlPagamento, a qual foi criada dentro do contexto do utilitário Pagamento, visando ao reúso.

Vale ressaltar que o diagrama da Figura 14 é apenas parcial, tendo em vista que apenas algumas classes de visão foram apresentadas. Para o projeto das classes de visão, optou-se por elaborar protótipos e mostrar apenas os layouts das classes mostradas na Figura 14. Assim, as figuras 15 a 19 mostram, respectivamente, os layouts das classes JanLocacao, JanDevolucao, JanCadastrarCliente, JanDadosTitular e JanDadosDependente³.

³ Observar que essas janelas não correspondem efetivamente ao projeto das janelas necessárias para o sistema da Videolocadora Passatempo. Elas são meramente ilustrativas de como seriam janelas deste tipo. Os layouts mostrados foram, na verdade, extraídos do sistema VideoLoc 1.4, disponível em <http://www.baixaki.com.br/download/videoloc.htm>.

Locação

Cliente:
João da Silva

☒ Titular ☐ Dependente

Financeiro

Última locação	Último pagamento	Saldo devedor
		0,00

Bloquear cliente

Filmes

Código: Título:

Código	Título	Prazo	Valor
00001	Meu Nome Não É Johnny	Sex	4,00

Itens: 1 Total: 4,00

Pagamento

Brinde
Remover
Nova data

Desconto Acréscimo

☐ Pago ☐ Pagou R\$

☒ A Pagar ☐ Imprimir recibo

Locar Cancelar

Figura 15 – Layout ilustrativo da janela JanLocacao

Devolução

Filme

Código: Título:
01 Meu Nome Não É Johnny

Cliente: João da Silva **Devolução:** 27/05/2010

Dados da locação

Data para dev.	Dias de atraso	Valor do filme	Locação paga
28/05/2010	0	4,00	Não

Lançar multa

☐ Imprimir comprovante de multa

Outros filmes com este cliente:

Código	Título	Prazo

Devolver Receber \$ Cancelar

Figura 16 – Layout ilustrativo da janela JanDevolucao

Cadastro de clientes

Nome: João da Silva Situação: Ativo Sexo: Masculino

Nascimento: 11/11/1971 RG: 454545 CPF: 452.452.452-52 Endereço: Av. Fernando Ferrari

Bairro: Goiabeiras CEP: 29060-900 Estado: ES Cidade:

Fone Residencial: Fone Comercial: Celular: E-Mail: joao.silva@gmail.com

Foto (FOTO)

Observação:

☐ Aceita receber e-mails sobre lançamentos e promoções

Gravar Cancelar

Figura 18 – Layout ilustrativo da janela JanDadosTitular

Cadastro de dependentes

Titular: João da Silva

Nome: João da Silva Filho Situação: Ativo

Parentesco: Filho (a) Telefone:

Foto (FOTO)

Observação:

Gravar Cancelar

Figura 19 – Layout ilustrativo da janela JanDadosDependente

5.3 – Camada de Gerência de Dados

No que se refere à camada de Gerência de Dados do pacote Atendimento à Cliente, abordagem análoga à adotada no pacote Controle de Acervo foi utilizada, ou seja, especialização do utilitário Persistência considerando as classes a serem persistidas, como mostra a Figura 20.

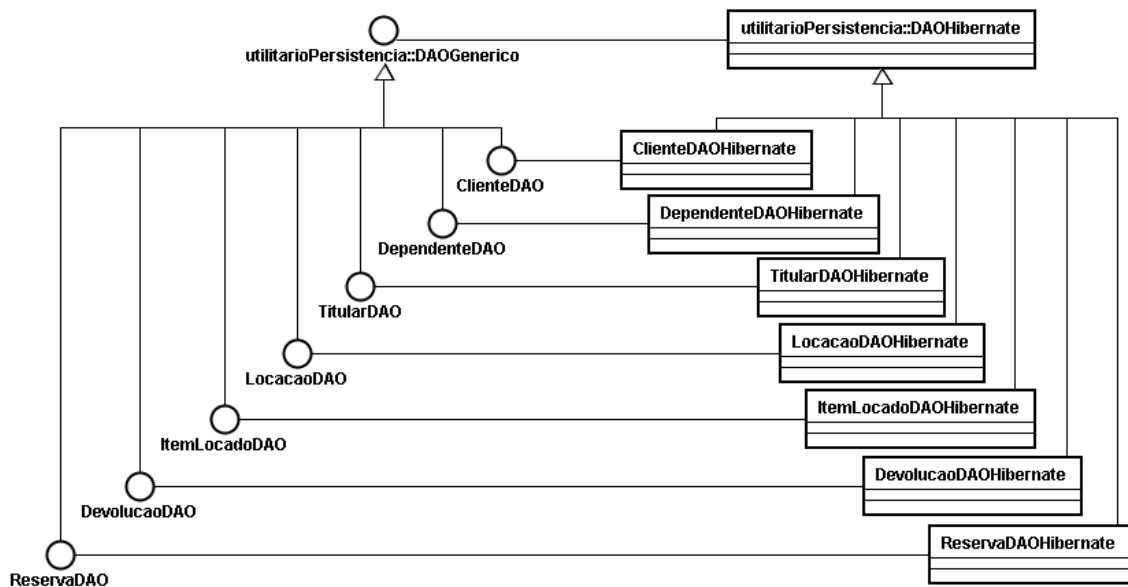


Figura 20 – CGD do pacote Atendimento à Cliente.