

5 Estudos de Caso

O primeiro estudo de caso apresentado é um sistema de sincronização de dados. Ele foi escolhido porque é totalmente voltado para a camada de persistência. O segundo estudo de caso é um site de leilão virtual, e o mesmo foi escolhido como caso de estudo por não utilizar um framework de persistência, ficando a cargo do processo proposto realizar a adaptação do framework.

5.1. Sistema de Sincronização dos Dados Acadêmicos do SAU e do AulaNet

O primeiro caso de estudo foi apresentado como um projeto final do curso de Bacharelado em Informática da Puc-Rio. Ele tem como objetivo sincronizar a base de dados do AulaNet com os dados acadêmicos da base de dados do SAU (Sistema de Administração Universitária).

Diversas disciplinas de graduação e de pós-graduação da PUC-Rio são ministradas parcialmente à distância utilizando o AulaNet [AulaNet]. É desejo da Universidade oferecer também disciplinas totalmente à distância, motivada pela nova regulamentação do Ministério da Educação (MEC). Entretanto, os processos acadêmicos voltados à admissão e manutenção da vida escolar do aluno, administrados pelo SAU - como criação de disciplinas, matrícula de alunos, entre outros -, têm que ser repetidos no AulaNet de forma manual, gerando possíveis problemas de inconsistência. O objetivo do sistema proposto é automatizar esses processos, através da sincronização da base de dados do AulaNet a partir da base de dados do SAU.

Esse estudo de caso foi utilizado por ser totalmente voltado para a sincronização de bases de dados, ou seja, é um sistema fundamentalmente voltado para a parte de persistência. Por este motivo será possível demonstrar qual seria o ganho ao desenvolver o mesmo utilizando o processo proposto nesta dissertação.

A primeira etapa do processo é a de criação de modelos independentes de plataforma (PIM) utilizando a linguagem UML. Nesta tarefa foi utilizada a ferramenta CASE Poseidon [Poseidon], tanto para a criação dos diagramas UML quanto para a geração dos arquivos XMI.

Abaixo a figura 40, mostra o modelo PIM criado a partir do estudo de caso em questão, para ser utilizado no processo.

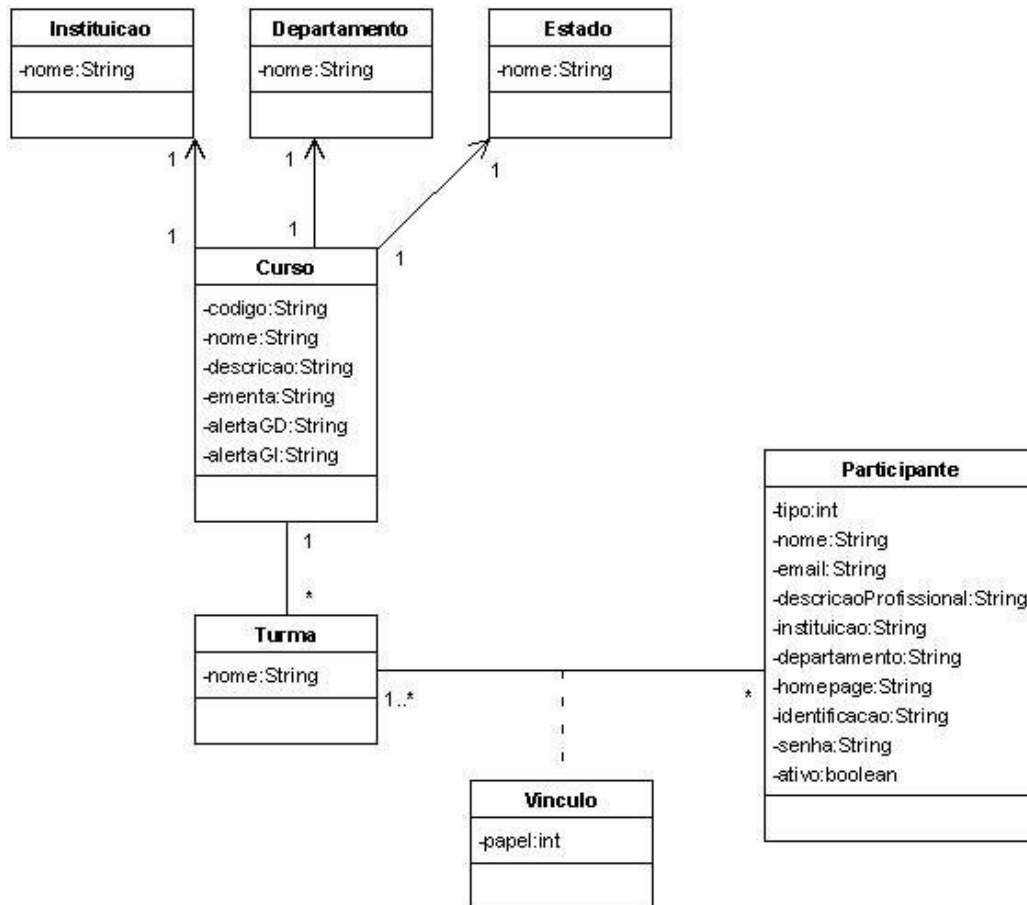


Figura 40 - Modelo PIM do Sistema de Sincronização dos Dados Acadêmicos do SAU e do AulaNet

Após a criação do modelo PIM é necessário configurar o mesmo, incluindo as marcações em forma de esteriótipos. Essas marcações estão diretamente ligadas às camadas previamente cadastradas na base de conhecimento para a camada a ser adaptada. Os esteriótipos configurados no modelo devem ser exatamente os mesmos que constam na base de conhecimento. Abaixo a figura 41 mostra o modelo PIM configurado.

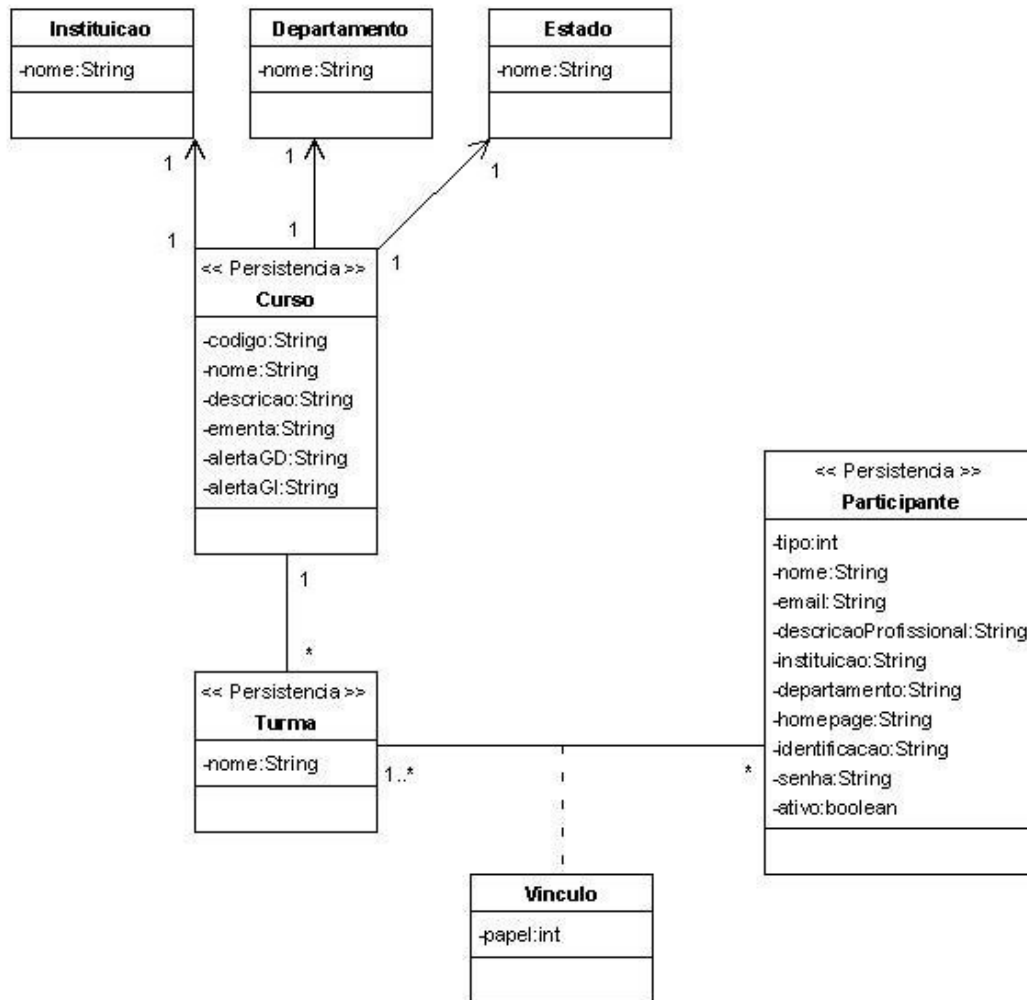


Figura 41 - Modelo PIM Configurado para a Camada de Persistência

O modelo PIM configurado agora é exportado, através da ferramenta CASE, para um arquivo no formato XML, para que o mesmo possa ser importado pela ferramenta de apoio ao processo.

Na ferramenta é necessário escolher as tecnologias que serão utilizadas para a transformação do modelo PIM no modelo PSM. Nesse caso iremos utilizar o padrão DAO (Data Access Object) [Gamma99] e o padrão Facade [Gamma99], pois o caso de estudo em questão utiliza os mesmos.

Para utilizar o padrão DAO, será necessário preencher algumas lacunas relacionadas às regras de transformações do mesmo. A figura 42 mostra o arquivo RDL e os dados necessários para que o mesmo esteja completo.

```

COOBOOK Example
RECIPE main
varDAOFactoryPai = NEW_CLASS ("DAOFactory");

LOOP
// PARA CADA DAO FACTORY
varNomeDAOFactory = ?;
varDAOFactory = NEW_CLASS (varNomeDAOFactory + "DAOFactory");
NEW_INHERITANCE (varDAOFactoryPai, varDAOFactory);

LOOP
// PARA CADA DAO
varNomeDAO = ?;
varDAO = NEW_CLASS (varNomeDAOFactory + varNomeDAO + "DAO");

NEW_METHOD (varDAO, "selecionar" + varNomeDAO);
NEW_METHOD (varDAO, "selecionar" + varNomeDAO + "s");
NEW_METHOD (varDAO, "inserir" + varNomeDAO);
NEW_METHOD (varDAO, "alterar" + varNomeDAO);
NEW_METHOD (varDAO, "excluir" + varNomeDAO);

// PARA CADA INTERFACE
varInterfaceDAO = NEW_INTERFACE (varNomeDAO + "DAO");
INTERFACE_IMPLEMENTATION (varDAO, varInterfaceDAO);

NEW_METHOD (varInterfaceDAO, "selecionar" + varNomeDAO);
NEW_METHOD (varInterfaceDAO, "selecionar" + varNomeDAO + "s");
NEW_METHOD (varInterfaceDAO, "inserir" + varNomeDAO);
NEW_METHOD (varInterfaceDAO, "alterar" + varNomeDAO);
NEW_METHOD (varInterfaceDAO, "excluir" + varNomeDAO);

// ADICIONA NO DAO FACTORY PAI
NEW_METHOD (varDAOFactoryPai, "get" + varNomeDAO + "DAO");
LOOP;
LOOP;
END_RECIPE;
END_COOKBOOK;

```

É necessário informar o nome de cada DAO Factory

É necessário informar o nome de cada DAO

Figura 42 - Arquivo RDL do Padrão DAO

Segundo a figura 5.3 acima é necessário informar um nome para cada DAOFactory e um nome para cada DAO de cada DAOFactory. De acordo com o estudo de caso em questão, seria necessário informar AULANET para a primeira DAOFactory e Participante, Turma e Curso para os DAOs da mesma. A segunda DAOFactory deve se chamar SAU, e cada uma dos DAOs se chamariam Participante, Turma e Curso, respectivamente. Após a inclusão desses dados, a máquina virtual RDL poderá executar o mesmo e atualizar o arquivo XML de entrada.

Agora o próximo passo é a configuração do padrão FACADE, onde também será necessário que algumas informações sejam passadas ao mesmo. A figura 43 mostra o arquivo RDL e os dados necessários para que o mesmo fique completo.

```
COOBOOK Example
RECIPE main
  varFachadaClasse = NEW_CLASS (?);

  LOOP
    // PARA CADA CLASSE
    varNomeClasse = ?;
    NEW_METHOD (varFachadaClasse, "selecionar" + varNomeClasse);
    NEW_METHOD (varFachadaClasse, "selecionar" + varNomeClasse + "s");
    NEW_METHOD (varFachadaClasse, "inserir" + varNomeClasse);
    NEW_METHOD (varFachadaClasse, "alterar" + varNomeClasse);
    NEW_METHOD (varFachadaClasse, "excluir" + varNomeClasse);
  LOOP;
END_RECIPE;
END_COOKBOOK;
```

É necessário informar o nome da Fachada

É necessário informar o nome de cada Classe

Figura 43 - Arquivo RDL do Padrão Facade

Ainda segundo o caso de estudo, será necessário informar o nome FachadaPersistencia para o nome da Fachada e TurmaAulaNet, CursoAulaNet e ParticipanteAulaNet, para serem adicionados na forma de métodos na Fachada. Após a inclusão desses dados, a máquina virtual RDL poderá executar o mesmo e atualizar o arquivo XML de entrada.

O arquivo XML de entrada está agora atualizado com o padrão DAO e FACADE. Sendo assim, será possível exportar o mesmo como um modelo PSM.

Após o arquivo XML de saída ser gerado, o mesmo deverá ser importado em uma ferramenta CASE, para que possa ser visualizado. A figura 44 abaixo mostra o modelo PSM gerado pelo processo proposto e importado pela ferramenta CASE Poseidon.

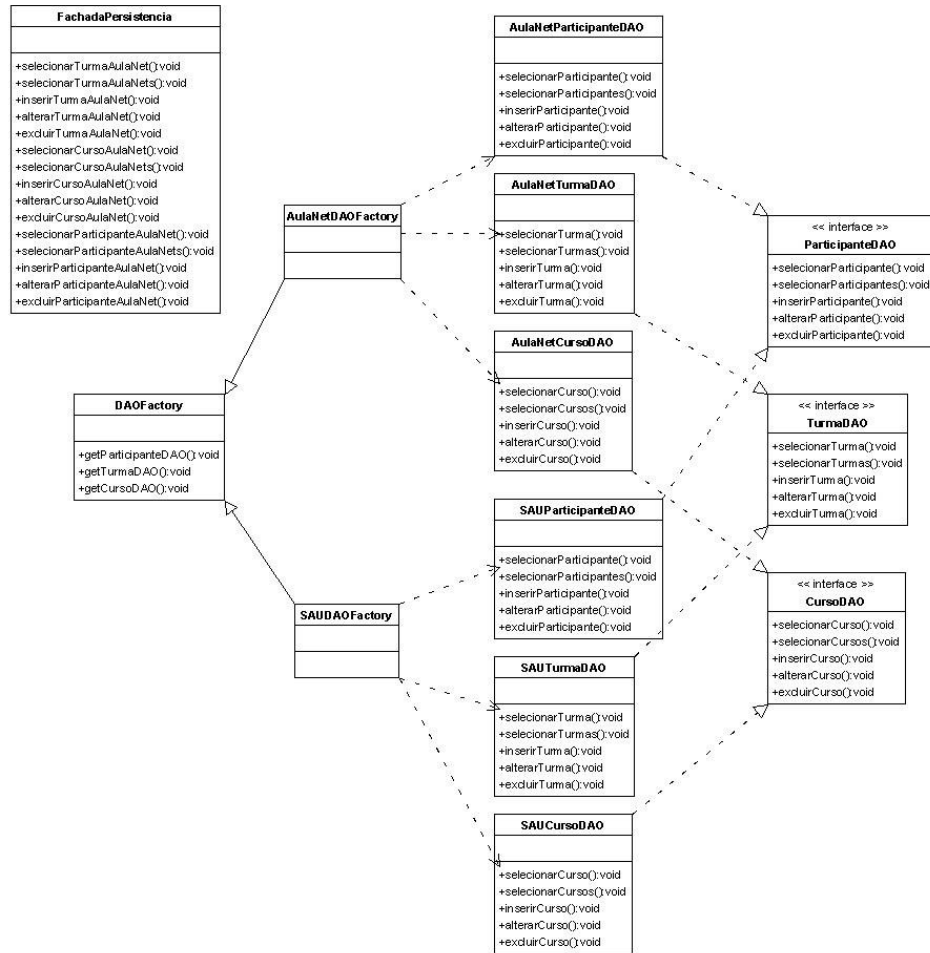
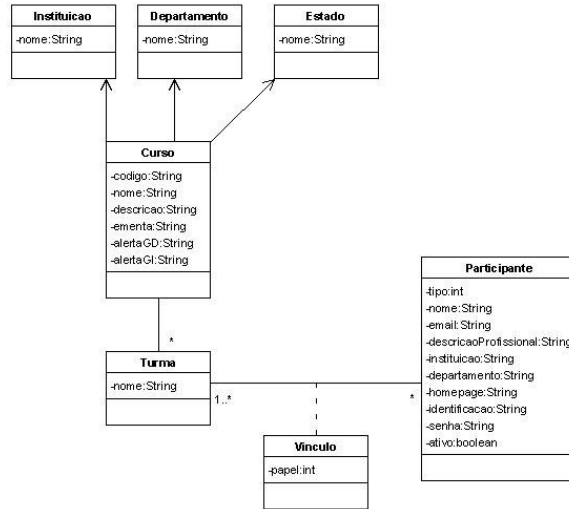


Figura 44 - Modelo PSM Gerado

Na figura 45 abaixo, é possível visualizar o modelo original (modelo já existente) com as setas indicando o que seria necessário adicionar manualmente ao modelo original para obtermos o modelo gerado pelo processo aqui descrito. A grande diferença entre os modelos está nos métodos de negócio de algumas classes, onde os mesmos teriam que ser adicionados no modelo gerado pelo processo. A estrutura dos padrões, incluindo relacionamentos, heranças, entre outros, foram gerados automaticamente. Mesmo em relação aos métodos, a grande maioria foi gerada.

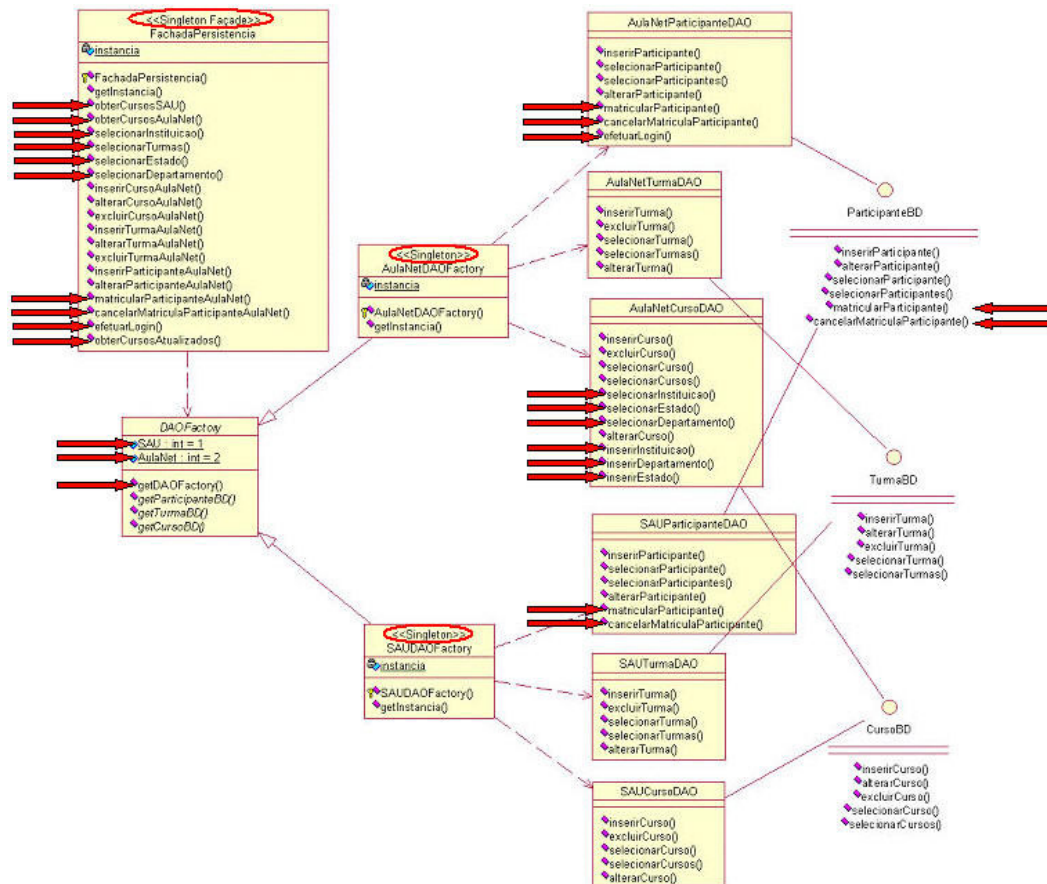


Figura 45 - Comparação do Modelo Original com o Modelo Gerado pelo Processo

5.2. Site de Leilão Virtual

Desenvolvido com o intuito de prover serviços de comércio eletrônico, mais especificamente de um leilão virtual, o sistema põe a disposição de seus

usuários toda a parte de compra, venda e troca de produtos. O mesmo também fornece o acompanhamento necessário as negociações ativas, seja pelo próprio site, como por email.

Esse sistema foi escolhido como caso de estudo por não utilizar mecanismos de persistência. Então será possível avaliar quais os prós e contras de se utilizar o processo proposto na adaptação de mecanismos de persistência em sistemas previamente prontos.

Como esse sistema não foi desenvolvido utilizando a abordagem MDA, será necessário criar um modelo PIM para o mesmo. Esse modelo será baseado no atual, com as devidas mudanças feitas. O objetivo é criar um modelo alto nível que será utilizado na geração dos modelos PSMs, adaptados a mecanismos de persistência.

Abaixo a figura 46 mostra o modelo atual do sistema, sendo que somente a parte de negócio, que será utilizada como o modelo PIM para posteriormente gerar os modelos PSM.

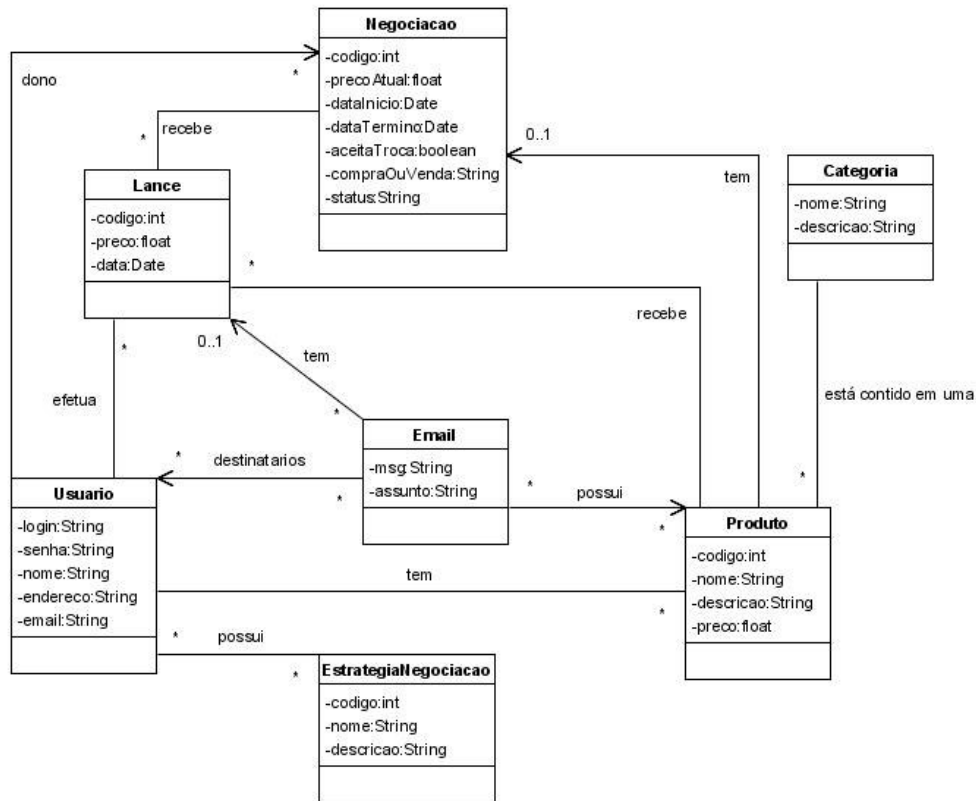


Figura 46 - Modelo PIM do Site de Leilão Virtual

Após a criação do modelo PIM é necessário configurar o mesmo, esta etapa é feita pelo Model Designer através de marcações indicando que camada e quais classes serão adaptadas. A figura 47 abaixo mostra o modelo PIM configurado pelo Model Designer.

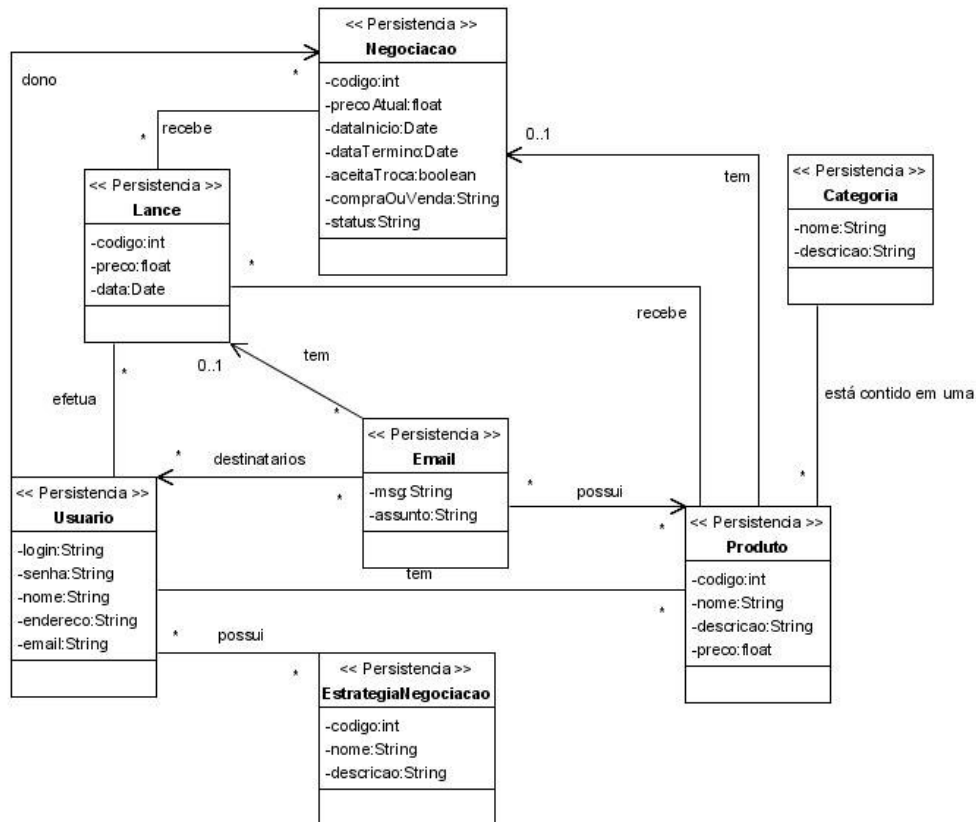


Figura 47 - Modelo PIM Configurado para a Camada de Persistência

O mecanismo de persistência Hibernate será utilizado durante o processo, mas o mesmo não consta na base de conhecimento, com isso o Knowledge-Base Designer terá que cadastrá-lo para que o PSM Designer possa utilizá-lo durante o processo.

O mesmo deve criar um arquivo RDL contendo as regras de transformações do novo mecanismo de persistência a ser adicionado na base de conhecimento. A figura 48 ilustra o arquivo RDL que contém as regras de transformações do framework de persistência Hibernate.

```
COOKBOOK Hibernate
RECIPE main
  varSession = NEW_CLASS ("Session");
  varHibernateUtil = NEW_CLASS ("HibernateUtil");

  NEW_DEPENDENCY (varHibernateUtil, varSession);

  LOOP
    varNomeClasse = ?;
    varClasse = NEW_CLASS (varNomeClasse);
    NEW_DEPENDENCY (varSession, varClasse);
  END_LOOP;
END_RECIPE;
END_COOKBOOK;
```

Figura 48 - Arquivo RDL que contém as regras de transformações do framework de persistência Hibernate

O novo mecanismo deve ser agora adicionado na base de conhecimento, através da ferramenta de apoio. É através dela que o Knowledge-Base Designer irá informar a qual camada o mesmo pertence, e associar também o arquivo RDL correspondente as regras de transformações.

Após o Knowledge-Base Designer cadastrar o novo mecanismo de persistência, a execução do processo pode continuar e o PSM Designer pode escolher o Hibernate para ser adaptado ao modelo. A próxima vez que o framework Hibernate precisar ser utilizado, o mesmo não precisará ser cadastrado novamente. É adicionado também o padrão Facade para ser adaptado ao modelo.

O PSM Designer executa as transformações contidas nos arquivos RDL correspondentes, e o arquivo XMI de entrada é atualizado de acordo com as regras de transformação definidas nesses arquivos.

A figura 49 abaixo mostra o modelo PSM gerado através do processo proposto e importado na ferramenta CASE Poseidon.

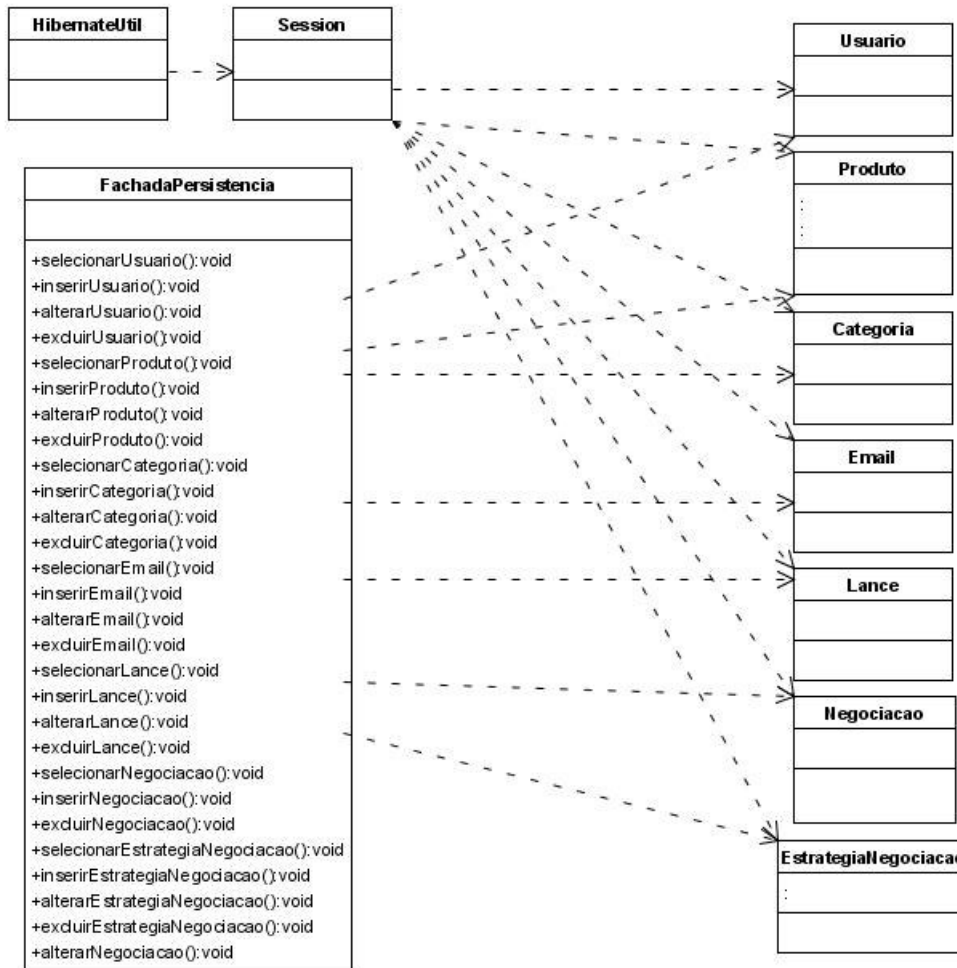


Figura 49 - Modelo PSM Gerado através do Processo Proposto e Importado na Ferramenta CASE Poseidon

O modelo gerado possui os métodos básicos de acesso a banco de cada objeto que continha a marcação “Persistência” no modelo PIM configurado. Foi incluído no modelo também o framework Hibernate, onde a classe Session é a responsável por manipular os objetos de negócio.

A dependência entre a fachada de persistência e o framework Hibernate devem ser incluídas pelo PSM Designer, pois as regras de transformação de cada tecnologia independe da outra.

A figura 50 abaixo mostra o modelo da camada de persistência finalizado pelo PSM Designer. As setas em vermelho indicam que o PSM Designer teve que incluir manualmente.

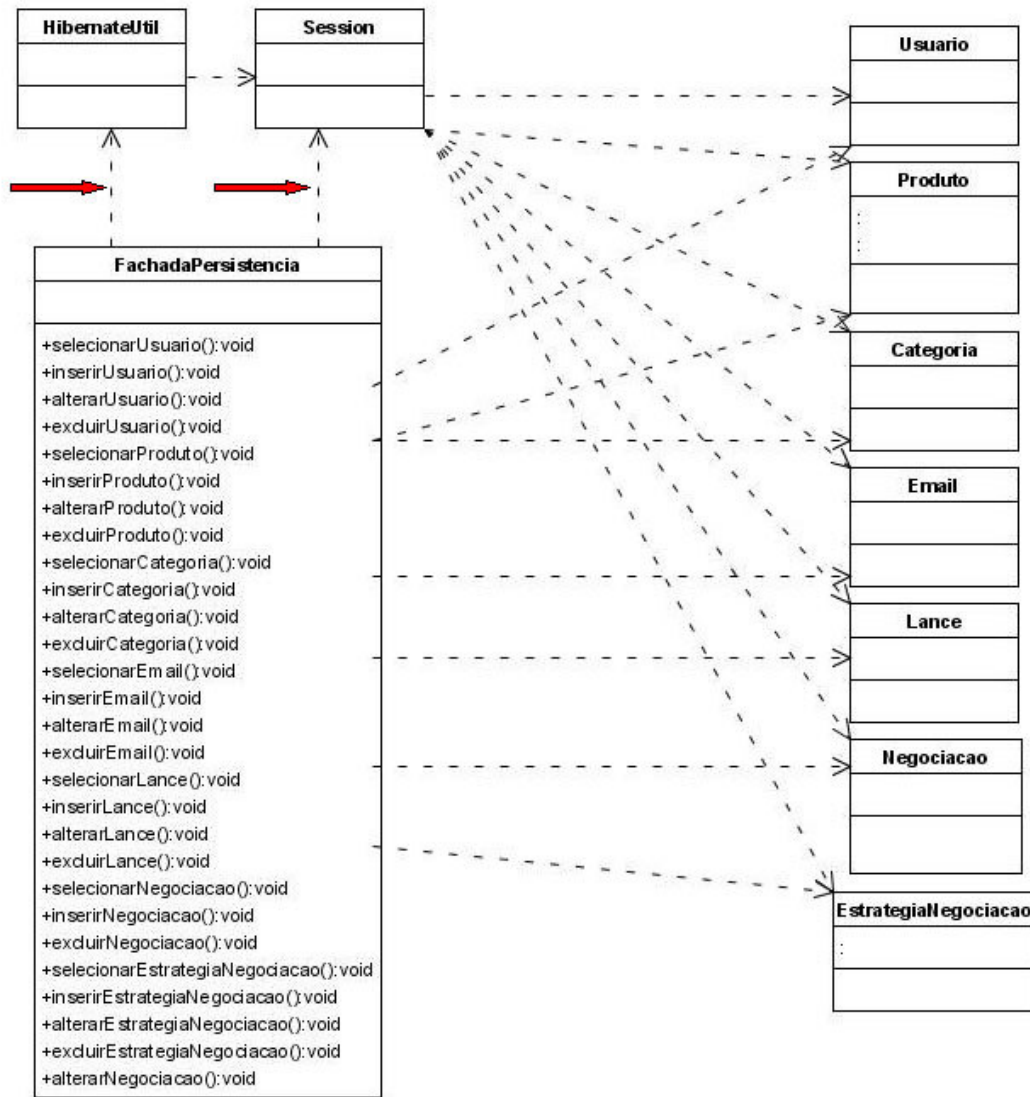


Figura 50 - Modelo da Camada de Persistência Finalizado pelo PSM Designer