



PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO DE JANEIRO
DEPARTAMENTO DE INFORMÁTICA
DISCIPLINA: PROJETO DE SISTEMAS DE SOFTWARE
PROF.º DR.º CARLOS LUCENA

GIVANILDE DE ASSIS DOS SANTOS

PROJETO DE SOFTWARE:
FREE RECOMMENDATION SYSTEM

JUNHO-2014

Lista de Ilustrações

Figura 1 - <i>Features</i> da versão <i>Basic</i>	3
Figura 2 - <i>Features</i> da versão <i>Advanced</i>	4
Figura 3 - <i>Features</i> da versão <i>Plus</i>	5
Figura 4 - <i>Uso do Padrão Singleton</i>	6
Figura 5 - Classe DAO<T> - Genérico	7
Figura 6 - Padrão MVC	7
Figura 7 - Classes do padrão <i>Filter</i>	8
Figura 8 - Padrão <i>Strategy</i>	8
Figura 9- Padrão <i>Mediator</i>	8
Figura 10 - Caso de Uso Geral	9
Figura 11 - Caso de uso da Versão <i>Basic</i>	10
Figura 12 - Casos de uso da versão <i>Advanced</i>	10
Figura 13 - Caso de uso da versão <i>Pro</i>	11
Figura 14 - Diagrama de Caso de usos Realiza Cadastro	12
Figura 15 - Diagrama de Caso de Comprador	13
Figura 16 - Diagrama de Caso do Vendedor	17
Figura 17 - Diagrama de Caso de Uso Administrador	19
Figura 18 - Diagrama de Caso de Uso Agentes	24
Figura 19 - Estrutura do Código Fonte	26
Figura 20 - Diagrama de Classes do Pacote DAO	26
Figura 21 - Diagrama de Classes do Pacote <i>Filter</i>	27
Figura 22 - Diagrama de Classes do Pacote MBean	27
Figura 23 - Diagrama de Classes do Pacote Util	28
Figura 24 - Diagrama de Classe do Pacote Modelo	28
Figura 25 - Diagrama de Classe do Pacote Agents Strategies	29
Figura 26 - Diagrama de Classe do Pacote Agents	29
Figura 27 - Diagrama de Sequência Cadastrar Usuário	30
Figura 28 - Diagrama de Sequência Acessar Sistema	30
Figura 29 - Diagrama de Sequência Atualiza dados pessoais	31
Figura 30 - Diagrama de Sequência Cadastra interesse	31
Figura 31 - Diagrama de Sequência Atualiza Status de Interesse	31
Figura 32 - Diagrama de Sequência Verificar Interesses	32
Figura 33 - Diagrama de Sequência Verificar Recomendações	32
Figura 34 - Diagrama de Sequência Atualiza Status de Recomendações	33
Figura 35 - Diagrama de Sequência Cadastrar Livros	33
Figura 36 - Diagrama de Sequência Verificar Livros	33
Figura 37 - Diagrama de Sequência Verificar Interessados	34
Figura 38 - Diagrama de Sequência Cadastrar Categoria	34
Figura 39 - Diagrama de Sequência Cadastrar Usuário Administrador	35
Figura 40 - Diagrama de Sequência Verificar Categorias	35
Figura 41 - Diagrama de Sequência Verificar Recomendação	36
Figura 42 - Diagrama de Sequência Enviar Email usuários sem Interesse	36
Figura 43 - Diagrama de Sequência Recomenda baseado no Interesse	37
Figura 44 - Diagrama de Sequência Recomenda baseado no Usuário	37
Figura 45 - Diagrama de Lógico de dados	38

Lista de Tabelas

Tabela 1- Descrição do UC01	12
Tabela 2 - Descrição do UC02.....	13
Tabela 3 - Descrição do UC03.....	14
Tabela 4 - Descrição do UC04.....	14
Tabela 5 - Descrição do UC05.....	15
Tabela 6 - Descrição do UC06.....	15
Tabela 7 - Descrição do UC07.....	16
Tabela 8 - Descrição do UC 08.....	16
Tabela 9 - Descrição do UC 09.....	17
Tabela 10 - Descrição do UC 10.....	18
Tabela 11 - Descrição do UC 11.....	18
Tabela 12 - Descrição do UC 12.....	19
Tabela 13 - Descrição do UC 13.....	20
Tabela 14 - Descrição do UC 14.....	20
Tabela 15 - Descrição do UC 15.....	21
Tabela 16 - Descrição do UC 16.....	21
Tabela 17 - Descrição do UC 17.....	22
Tabela 18 - Descrição do UC 18.....	22
Tabela 19 - Descrição do UC 19.....	23
Tabela 20 - Descrição do UC 20.....	24
Tabela 21 - Descrição do UC 21.....	25
Tabela 22 - Descrição do UC 22.....	25

Sumário

1	Introdução	2
1.1	Propósito	2
1.2	Problema de Pesquisa	2
2	Descrição do Problema	2
2.1	<i>Features - Versão Basic</i>	3
2.2	<i>Features - Versão Advanced</i>	4
2.3	<i>Features – Versão Plus</i>	5
3	Visão Geral da Linha de Produto	6
3.1	Padrões de Projeto utilizados	6
3.1.1	<i>Singleton</i>	6
3.1.2	<i>DAO</i>	6
3.1.3	<i>Model View Controller</i>	7
3.1.4	<i>Filter</i>	7
3.1.5	<i>Strategy</i>	8
3.1.6	<i>Mediator</i>	8
3.2	Descrição dos Agentes	9
3.3	Modelos	9
3.3.1	Análise do Domínio	9
3.3.1.1	Casos de Uso	9
3.3.2	Projeto do Domínio	26
3.3.2.1	Diagrama de Classes	26
3.3.2.1.1	Pacote DAO	26
3.3.2.1.2	Pacote Filter	27
3.3.2.1.3	Pacote MB	27
3.3.2.1.4	Pacote Util	28
3.3.2.1.5	Pacote Model	28
3.3.2.1.6	Pacote Agents	29
3.3.3	Diagramas de Sequência	29
3.3.3.1	Usuário Público	29
3.3.3.2	Usuário Comprador	30
3.3.3.3	Usuário Vendedor	33
3.3.3.4	Usuário Administrador	34
3.3.3.5	Usuário Agente	36
3.3.4	Projeto de Dados	38
4	Guia de Instalação e Configuração	38
4.1	Instalação no servidor web	38
4.1.1	Instalação na IDE	39
5	Anexo	40
5.1	Estruturas de Dados	40
5.2	Regras de Negócio	40

1 Introdução

O presente documento visa documentar e direcionar as atividades fundamentais ao desenvolvimento da linha de produto de software cujo objetivo é reunir em um ambiente web pessoas com interesse em comprar e vender livros.

Nesta seção será apresentada uma visão geral do presente documento, definindo-se o seu propósito no contexto do processo, o seu escopo de representação vinculado a Linha de Produto de Software em desenvolvimento, a visão geral do funcionamento do sistema bem como a maneira como as partes componentes deste artefato estão dispostas ao longo do presente texto.

1.1 Propósito

O propósito deste documento é fornecer um modelo de desenvolvimento que deve guiar as atividades de projeto e implementação do sistema *Free Recommendation System*. Neste sentido, o referido sistema propõe automatizar os processos inerentes ao cadastramentos de pessoas com interesses, no que diz respeito a compra ou venda de livros, bem como fornecer meios adequados para recomendar de modo efetivo estes desejos.

1.2 Problema de Pesquisa

O problema central desta pesquisa é desenvolver uma ferramenta que permita a partir de agentes inteligentes recomendar pessoas com interesses em adquirir produto (livro, música, filme) a possíveis vendedores e vice-versa.

2 Descrição do Problema

O *Free Recommendation System* é uma linha de produto desenvolvida com o propósito principal de propiciar o encontro de pessoas com interesses em comprar ou vender livros.

Nesta seção iremos descrever de maneira sucinta as versões existentes da linha de produto de software do presente trabalho. A fim de maiores detalhes, a respeito das funcionalidades de cada uma das versões, consulte a seção de casos de uso.

2.1 Features - Versão Basic

Esta é a primeira versão da Linha de Produto de Software. Ela basicamente contém todas as funcionalidades ligadas ao cadastramento, alteração e exclusão de dados e apresenta os dados de interesses/recomendação através de relatórios. De modo reduzido, esta versão da linha de produto representa a implementação dos CRUD¹s relativos aos dados do domínio da aplicação

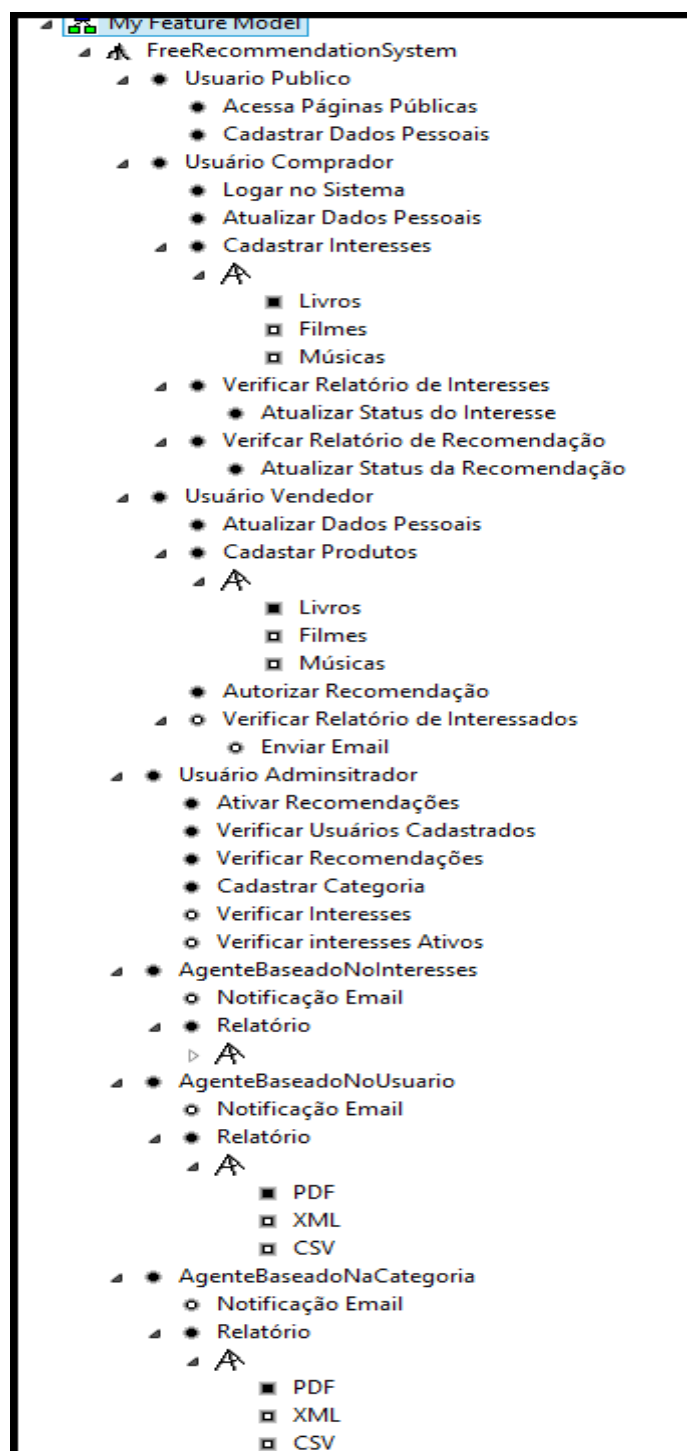


Figura 1 - Features da versão Basic

1 CRUD – Do inglês Create, Read, Update e delete que para as quatro operações básicas utilizadas em bancos de dados relacionais.

2.2 Features - Versão Advanced

Nesta versão da linha de produto, além de todas as *features* existente na *Basic*, a mesma acrescenta algumas funcionalidades para o usuário Administrador (verificar interesses e verificar interesses ativos), Vendedor (verificar relatório de interesses) e os Agentes (notificação por Email e por três tipos de relatórios).

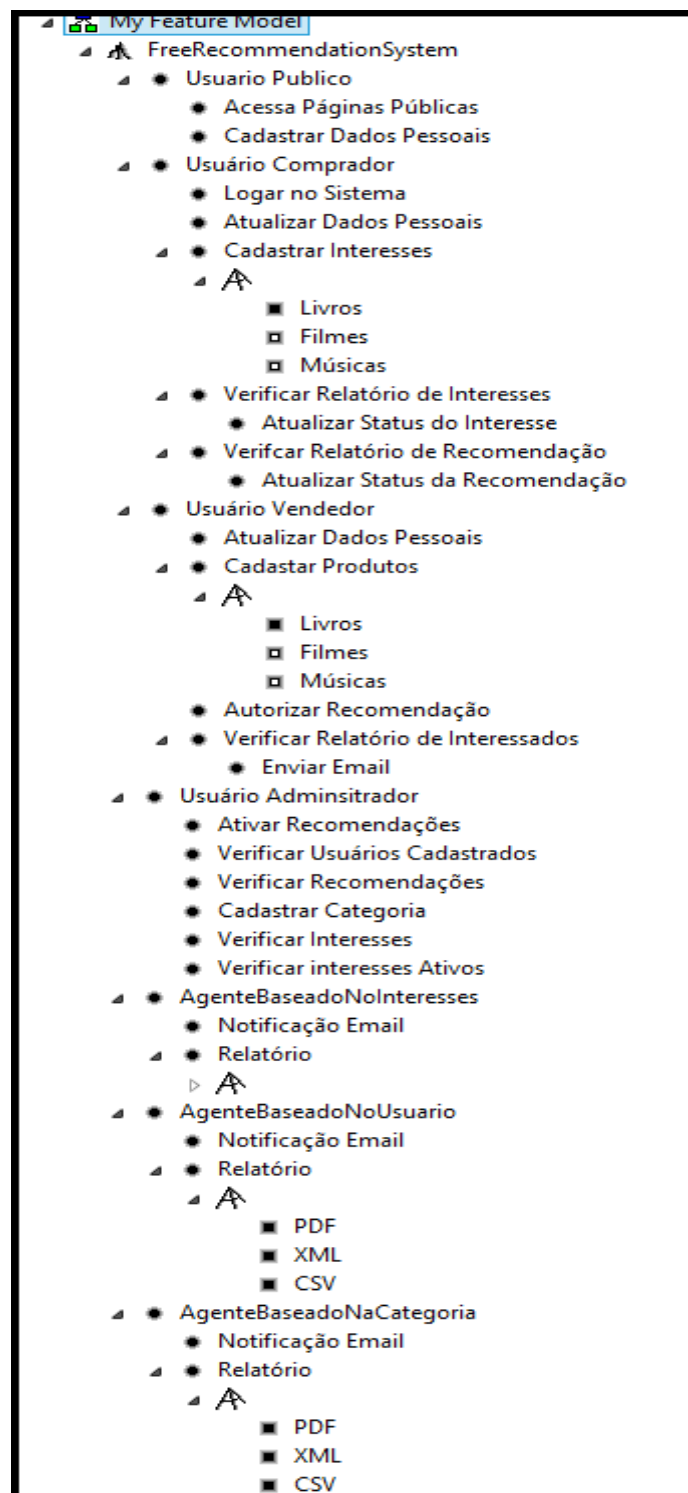


Figura 2 - *Features* da versão *Advanced*

2.3 Features – Versão Plus

Esta é a versão mais completa da linha de produto de software. Aqui serão apresentadas novas *features* em relação à versão anterior no que tange a inclusão de interesses relacionados a filmes e músicas.

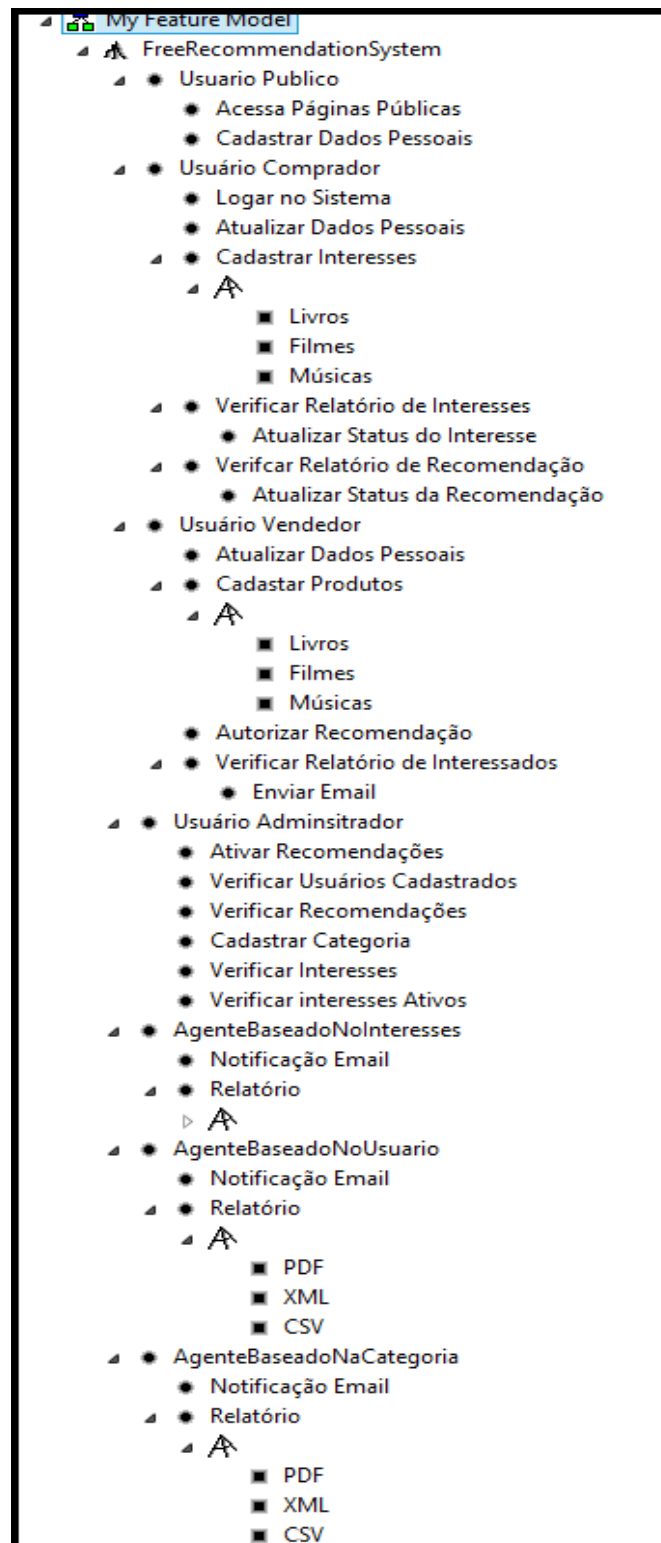


Figura 3 - Features da versão Plus

3 Visão Geral da Linha de Produto

3.1 Padrões de Projeto utilizados

Padrões de projeto descrevem uma solução geral reutilizável para um problema recorrente no desenvolvimento de sistemas de software orientados a objetos. Não é um código final, é uma descrição ou modelo de como resolver o problema do qual trata, que pode ser usada em muitas situações diferentes. Os Padrões de Projeto normalmente definem as relações e interações entre as classes ou objetos, sem especificar os detalhes das classes ou objetos envolvidos, ou seja, estão num nível de generalidade mais alto. Nesta seção serão apresentados os padrões de projetos que foram utilizados na construção do *Free Recommendation System*.

3.1.1 Singleton

Este padrão garante a existência de apenas uma instância de uma classe, mantendo um ponto global de acesso ao seu objeto. Portanto em relação a aplicação FRS, a classe *HibernateUtil*, precisa garantir que o *EntityManager* utilizado no sistema seja único, deste modo, o próprio *framework Object-Relational Mapping (ORM)* fornece esta facilidade objetivando garantir o bom funcionamento das transações realizadas no banco de dados.



Figura 4 - Uso do Padrão Singleton

3.1.2 DAO

O *Data Access Objects (DAO)* é um padrão para persistência de dados que permite separar regras de negócio das regras de acesso a banco de dados (BD). Deste modo, nas aplicações que fazem uso da arquitetura *Model View Controller (MVC)*, todas as funcionalidades de BD, tais como: obter as conexões, mapear objetos Java para tipos de dados SQL ou executar comandos *Structured Query Language (SQL)*, devem ser feitas por classes de DAO.

Assim, no FRS foi utilizado um objeto do tipo DAO que faz uso de *Generics Java* de maneira que o tipo particular a ser manipulado é injetado dinamicamente. Com isto implica-se em pouco esforço na criação de um DAO para cada modelo persistente do sistema.



Figura 5 - Classe DAO<T> - Genérico

3.1.3 Model View Controller

O padrão *Model View Controller* (MVC) um modelo de arquitetura de software que separa a representação da informação da interação do usuário com ele. Neste sentido temos:

- *Model* é consiste nos dados da aplicação, regras de negócios, lógica e funções.
- *View* é qualquer saída de representação dos dados, como uma tabela ou um diagrama..
- *Controller* faz a mediação da entrada, convertendo-a em comandos para o modelo ou visão.

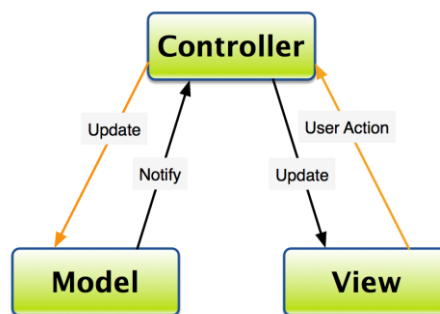


Figura 6 - Padrão MVC

Sendo assim, o *Free Recommendation System* foi implementado utilizando o JSF do J2EE o que facilitou a implementação e uso do MVC.

3.1.4 Filter

O padrão *Filter* é utilizado para análise e tratamento de dados. Sendo assim, através do mesmo é possível utilizar um ou mais objetos (filtros) que tratam ou operam um fluxo de dados de maneira específica, a fim de obter soluções genéricas (reutilizáveis). Neste sentido, o mesmo foi utilizado para determinar os tipos distintos de usuários que podem acessar o sistema.

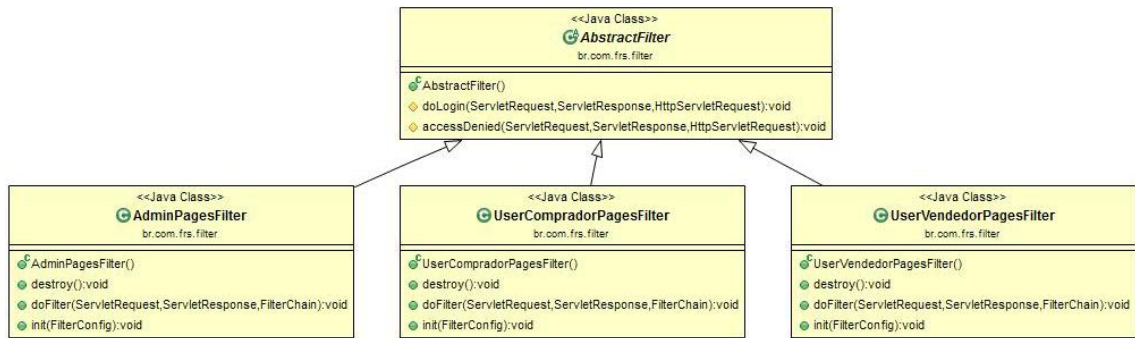


Figura 7 - Classes do padrão Filter

3.1.5 Strategy

O padrão *Strategy* tem como objetivo representar uma operação a ser realizada sobre os elementos de uma estrutura de objetos. Portanto, o mesmo permite definir novas operações sem alterar as classes dos elementos sobre os quais opera.

Assim, *Free Recommendation System* utiliza-se de diferentes formas de recomendações e geração de relatórios, o que motivou a utilização deste padrão.

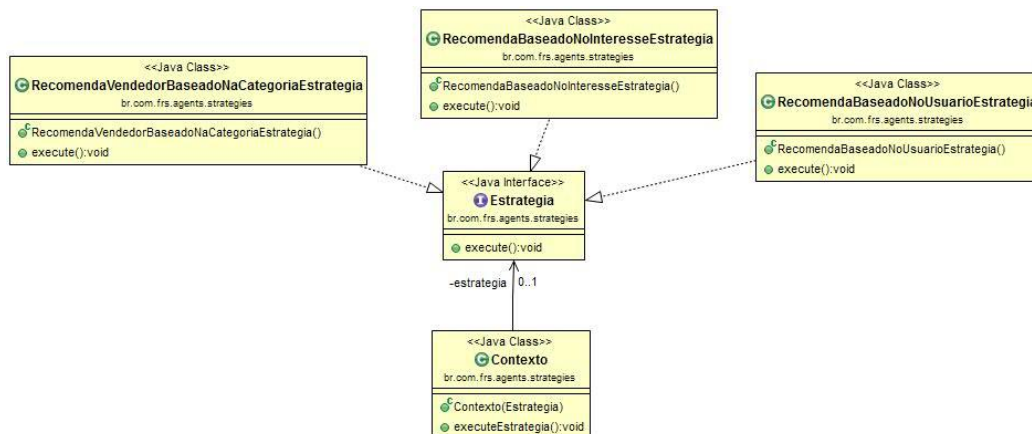


Figura 8 - Padrão Strategy

3.1.6 Mediator

Um *mediator*, é quem desacopla e gerencia as colaborações entre um grupo de objetos. Ou seja, define um objeto que encapsula as interações dentro desse grupo. Neste sentido a classe *AgentComunicator* é o responsável por fazer a comunicação entre os agentes.



Figura 9- Padrão Mediator

3.2 Descrição dos Agentes

O *Free Recommendation System* utiliza-se de três agentes:

- I. Baseado em Interesses, gera recomendações a partir de interesses cadastrados no sistema;
- II. Baseado no Usuário, gera recomendações a partir de interesses específicos de um Usuário;
- III. Envio de Email ao usuário sem interesse, alerta pessoas cadastradas no sistema que não possuem nenhum interesse associado;

3.3 Modelos

3.3.1 Análise do Domínio

3.3.1.1 Casos de Uso

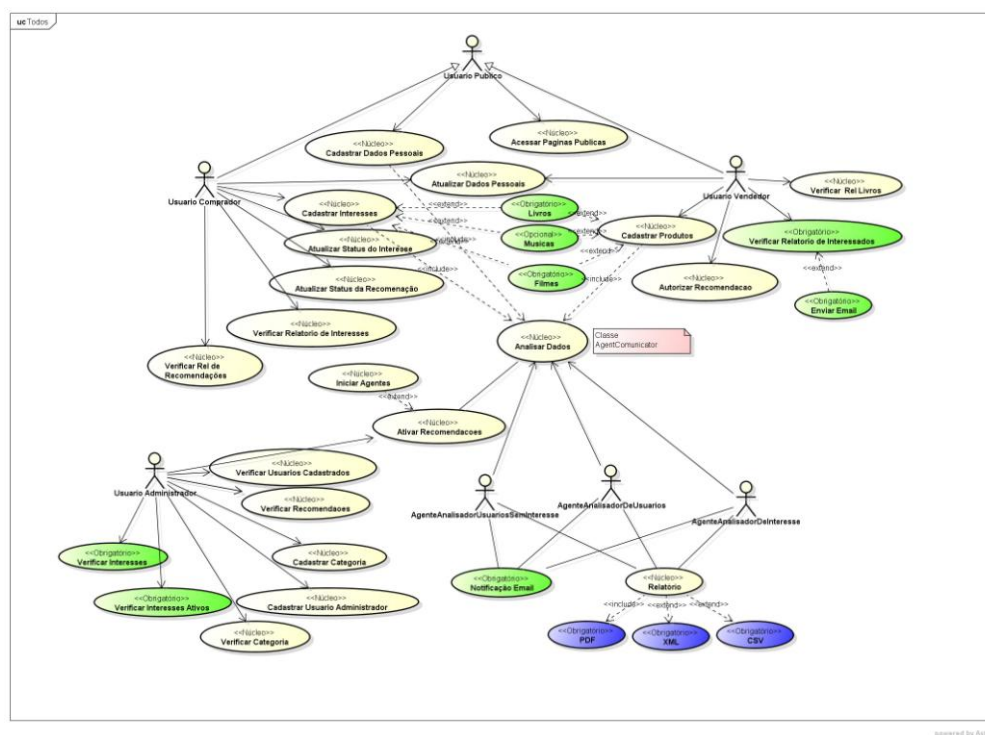
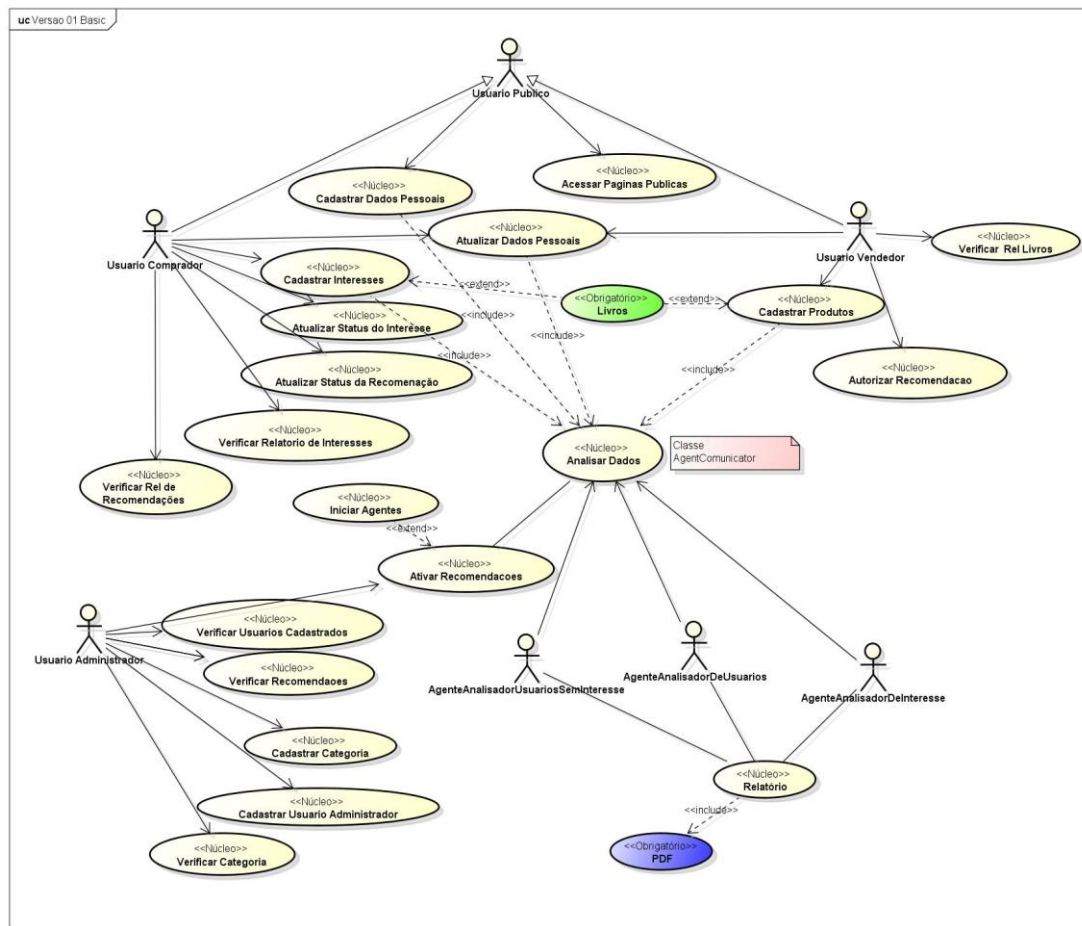


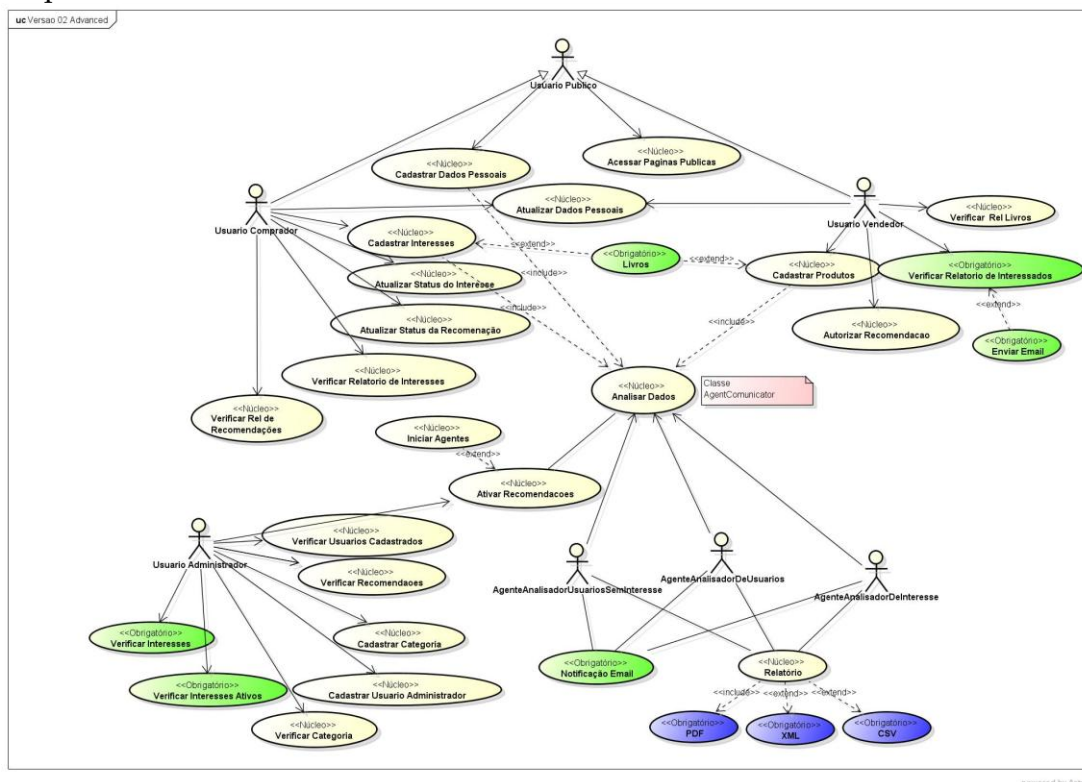
Figura 10 - Caso de Uso Geral

A **Figura 10** apresenta o núcleo do sistema e as possibilidades de variações através dos pacotes: OR e Alternativa. Deste modo, percebe-se que existem possibilidade de modificações na coleta de dados e na representação dos resultados para o usuário solicitante.

A fim de verificar as possíveis variações, as figuras subsequentes apresentam a versão *Basic*, *Advanced* e *Plus* da linha de produtos *Free Recommendation System*. Portanto a partir da **Figura 11** temos a visão da versão *Basic*.



A **Figura 12** apresenta a versão *Advanced*, neste sentido nota-se novas funcionalidades que não existiam na versão *Basic*.



A **Figura 13** apresenta a versão *Plus*, na qual a mesma possui todas as funcionalidades da linha da linha de produto.

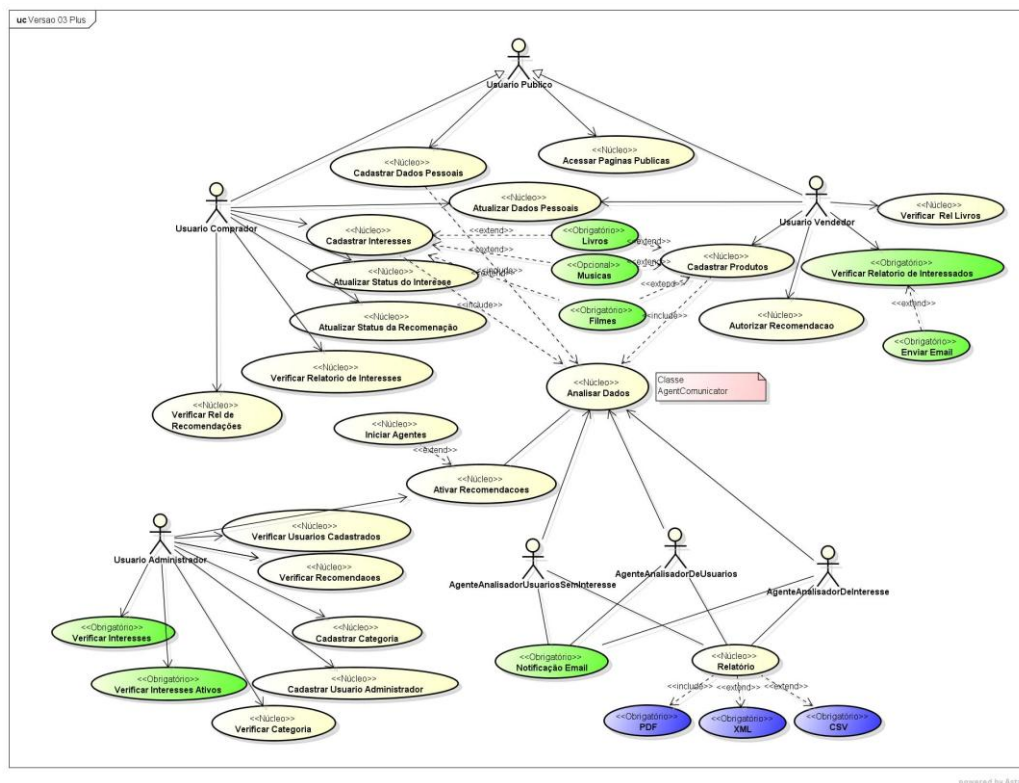
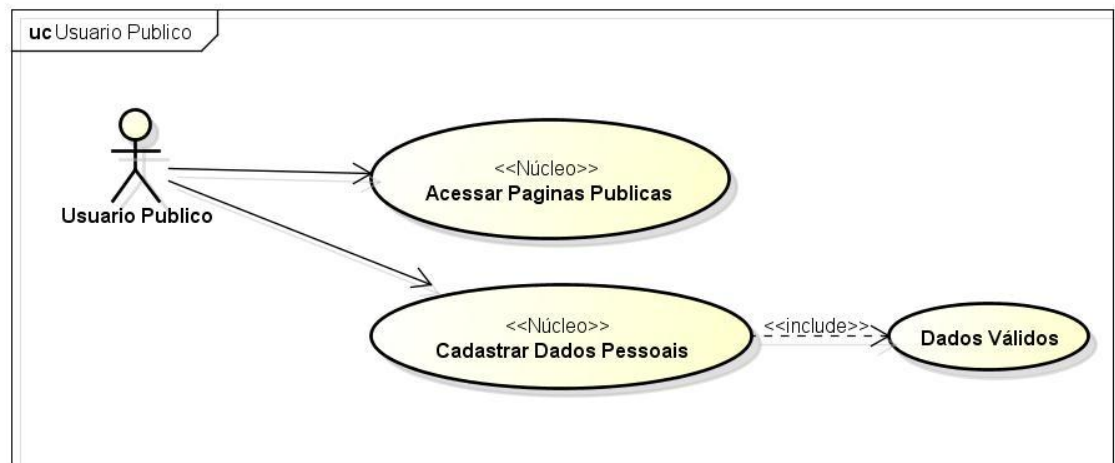


Figura 13 - Caso de uso da versão *Pro*

A seguir serão detalhados os casos de usos, salienta-se que alguns dos diagramas não foram detalhados visto que o processo é o mesmo, e a única modificação ocorre no tipo de ator, neste sentido os mesmos foram desconsiderados neste documento.



powered by Astah

Figura 14 - Diagrama de Caso de usos Realiza Cadastro

Tabela 1- Descrição do UC01

UC 01 – Cadastrar Usuário
Descrição sucinta – Caso de uso utilizado para cadastro de um usuário no sistema
Atores – Usuário público
Pré-condições – Acessar a página /FRS-PUC/public/usuario.jsf
Pós-condições – Não se aplica.
Fluxo básico:
1- O usuário acessa a página
2- O sistema mostra um formulário
3- O usuário preenche um formulário.
4- O usuário clica em gravar respeitando a validação do formulário (RN1)
5- O usuário é redirecionado para a página FRS-PUC /public/login.jsf
Fluxos alternativos:
1- O sistema exibe a mensagem de validação
2- Retorna ao passo 1 do fluxo principal.
Fluxos de exceção – Violação da RN1 e feedback ao usuário do motivo.
Estruturas de dados – ED1 e ED2
Regras de negócio – RN1
Observações - Não se aplica.

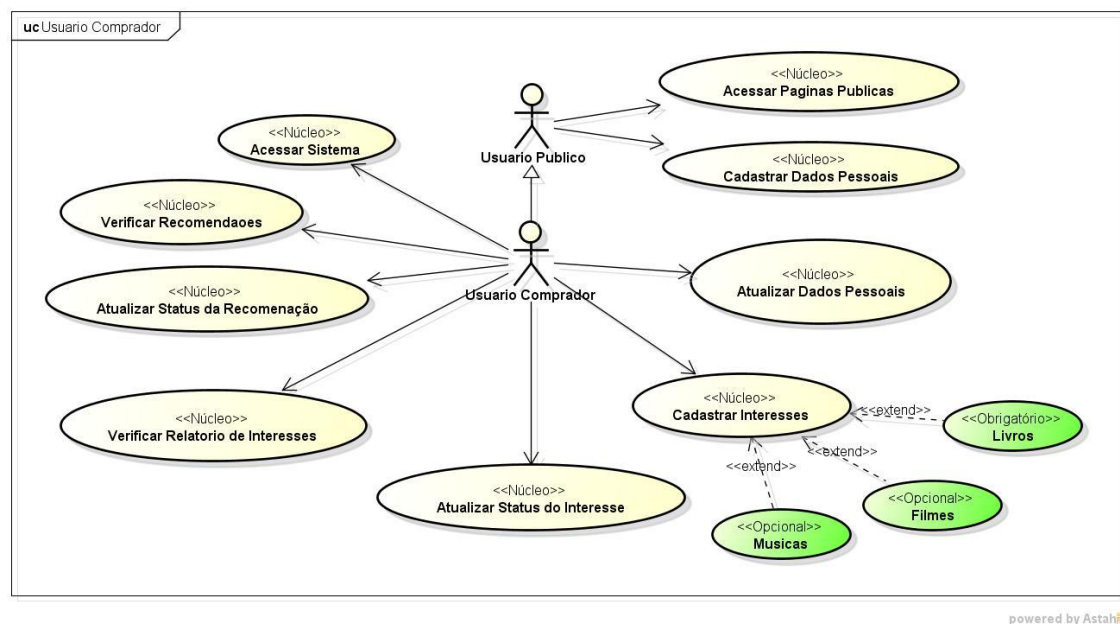


Figura 15 - Diagrama de Caso de Comprador

Tabela 2 - Descrição do UC02

UC 02 – Acessar sistema
Descrição sucinta – Utilizado para realização do Login e acesso ao sistema.
Atores - Comprador/Vendedor/Administrador
Pré-condições – Realizar o cadastro no sistema e acessar a página login.jsf
Pós-condições – Acessar a página padrão de acordo com a role do usuário
Fluxo básico:
1- Acessar login.jsf
2- Será mostrado um formulário
3- Preencher o formulário com login e senha
4- Redirecionamento para a página padrão (de acordo com o usuário)
5- Fim do caso de uso
Fluxos alternativos:
3- O sistema exibe a mensagem de validação
4- Retorna ao passo 1 do fluxo principal.
Fluxos de exceção – Violação da RN2
Estruturas de dados – ED1 e ED2
Regras de negócio – RN2
Observações

Tabela 3 - Descrição do UC03

UC 03 – Cadastrar interesse
Descrição sucinta: Cadastrar interesses
Atores: Comprador
Pré-condições: Acessar sistema / possuir perfil de usuarioComprador
Pós-condições: Ver reserva na página relInteresse.jsf
Fluxo básico:
1- Acessar o sistema
2- Realizar login com role de usuário comprador
3- Acessar interesse.jsf
4- Será mostrado um formulário para cadastro
5- Preencher o formulário
6- Receber a mensagem de confirmação do cadastro
7- Fim do caso de uso
Fluxos alternativos:
5- Erro na entrada de dados
6- Retorno ao passo 5 do fluxo anterior.
Fluxos de exceção – Violação da RN3
Estruturas de dados – ED3 e ED4
Regras de negócio – RN5 e RN6

Tabela 4 - Descrição do UC04

UC 04 – Atualizar dados pessoais
Descrição sucinta: Verificação dos dados e/ou atualizar
Atores: Comprador/vendedor/administrado
Pré-condições: Acessar sistema / possuir role / acessar pagina relDadosUser.jsf
Pós-condições: Verificar alteração dos dados
Fluxo básico:
1- Acessar o sistema e realizar login
2- Acessar a pagina relDadosUser.jsf
3- Verificar os dados
4- Fim do caso de uso
Fluxos alternativos:
1- Alterar os dados em algum campo do formulário
2- Voltar a etapa 3 do fluxo principal
Fluxos de exceção: Não se aplica.
Estruturas de dados: ED1
Regras de negócio: RN4 e RN5
Observações

Tabela 5 - Descrição do UC05

UC 05 – Atualizar status de Interesses
Descrição sucinta: Atualizar o status dos interesses cadastrados no sistema
Atores: Comprador
Pré-condições: Acessar como usuarioComprador / acessar pagina relInteresse.jsf
Pós-condições: Não se aplica
Fluxo básico:
1- Acessar o sistema
2 - Realizar login com perfil de usuário comprador
3- Acessar a pagina usuarioComprador/relInteresse.jsf
4- Escolher um novo status para o interesse
5- Verificar mudança de <i>status</i>
6- Fim do caso de uso
Fluxos alternativos: Inexistência de interesses cadastrados
Fluxos de exceção: Problemas com o gerenciamento do <i>Hibernate</i>
Estruturas de dados: ED1, ED5 e ED6
Regras de negócio: RN9
Observações

Tabela 6 - Descrição do UC06

UC 06 – Verificar Relatório de Interesses
Descrição sucinta: Verificação de todos os interesses cadastrados no sistema
Atores: Comprador
Pré-condições: Acessar sistema como usuarioComprador e acessar relInteresses.jsf
Pós-condições: Não se aplica
Fluxo básico:
1- Acessar o sistema
2- Realizar login com role de usuário comprador
3- Acessar a pagina usuarioComprador/relatórios/relInteresses.jsf
4- Verificar interesses cadastrados
5- Fim do caso de uso
Fluxos alternativos: Inexistência de interesses cadastrados
Fluxo de exceção - Não se aplica
Estruturas de dados: ED1, ED5 e ED6
Regras de negócio: RN7
Observações

Tabela 7 - Descrição do UC07

UC 07 – Verificar Recomendação
Descrição sucinta: Verificação as recomendações cadastrados no sistema
Atores: Comprador
Pré-condições: Acessar sistema como usuarioComprador / acessar pagina relRecomendacoesComprador.jsf
Pós-condições: Não se aplica
Fluxo básico: 1- Acessar o sistema 2- Realizar login com role de usuário comprador 3- Acessar a pagina usuarioComprador/relatórios/relRecomendacoesComprador.jsf 4- Verificar as recomendações cadastradas 5- Fim do caso de uso
Fluxos alternativos: Inexistência de recomendações cadastrados
Fluxos de exceção: Não se aplica
Estruturas de dados: ED1, ED7 e ED8
Regras de negócio: RN10 e RN11
Observações

Tabela 8 - Descrição do UC 08

UC 08 – Atualizar Status da Recomendação
Descrição sucinta: Atualizar o status dos interesses cadastrados no sistema
Atores: Comprador
Pré-condições: Acessar sistema como usuarioComprador / acessar pagina relRecomendacoesComprador.jsf
Pós-condições: Não se aplica
Fluxo básico: 1 - Acessar o sistema 2 - Realizar login com role de usuário comprador 3 - Acessar a pagina usuarioComprador/relatórios/relRecomendacoesComprador.jsf 4 - Escolher um novo status para recomendação 5 - Verificar os dados 6 - Fim do caso de uso
Fluxos alternativos: Quando uma recomendação é alterada para confirmada, um email será enviado para o proprietário do livro informando o interesse do usuário comprador
Fluxos de exceção: Problemas com o gerenciamento do <i>Hibernate</i>
Estruturas de dados: ED1, ED7 e ED8
Regras de negócio: RN10 e RN11
Observações

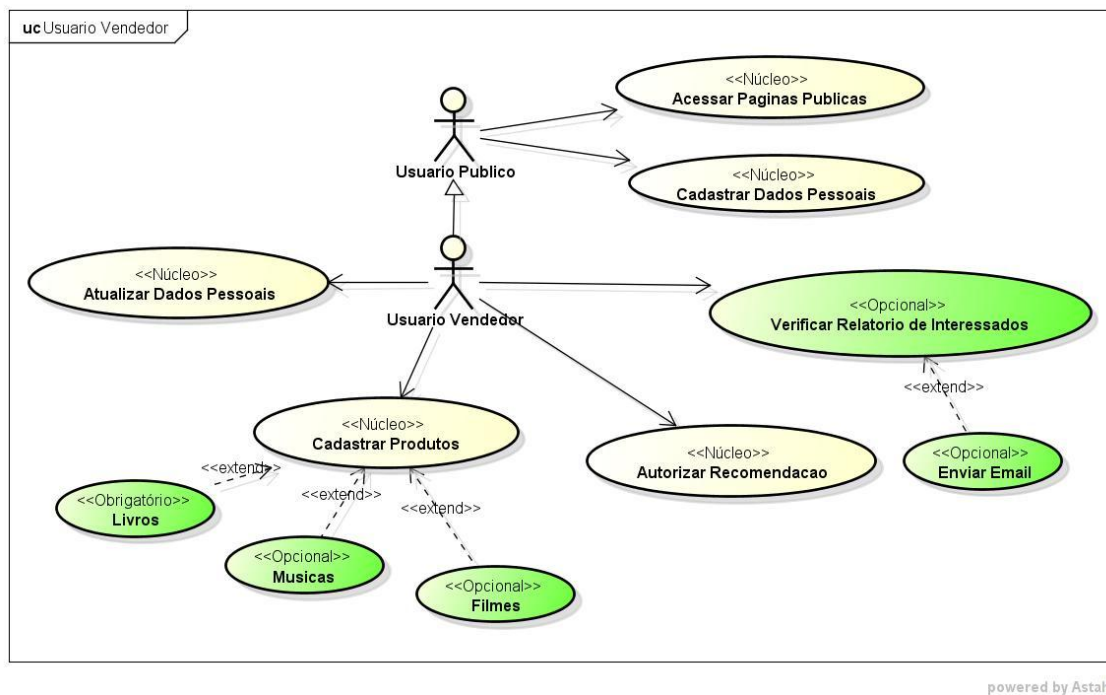


Figura 16 - Diagrama de Caso do Vendedor

Tabela 9 - Descrição do UC 09

UC 09 – Cadastrar Produtos
Descrição sucinta: Cadastrar livros para disponibilizar para venda
Atores: Vendedor
Pré-condições: Acessar sistema como usuarioVendedor / acessar livro.jsf
Pós-condições: Não se aplica
Fluxo básico:
1- Acessar o sistema
2- Realizar login com role de usuário vendedor
3- Acessar livro.jsf
4- Será mostrado um formulário para cadastro
5- Preencher o formulário
6- Receber a mensagem de confirmação do cadastro
7- Fim do caso de uso
Fluxos alternativos:
5- Erro na entrada de dados
6- Retorno ao passo 5 do fluxo anterior.
Fluxos de exceção – Violação da RN3
Estruturas de dados – ED3 e ED4
Regras de negócio – RN3

Tabela 10 - Descrição do UC 10

UC 10 – Verificar Relatório de Interessados
Descrição sucinta: Verificação das pessoas interessadas em um livro compatível com a categoria dos seus.
Atores: Vendedor
Pré-condições: Acessar sistema como vendedor / acessar pagina relInteressesVendedor.jsf
Pós-condições: Não se aplica
Fluxo básico: 1- Acessar o sistema 2- Realizar login com role de usuário vendedor 3- Acessar a pagina usuarioVendedor/relatórios/relInteressesVendedor.jsf 4- Verificar pessoas interessadas em seus livros 5- Fim do caso de uso
Fluxos alternativos: Inexistência de pessoas interesses nos livros
Fluxos de exceção: Não se aplica.
Estruturas de Dados: ED1, ED3 e ED4
Regras de negócios: RN4
Observações

Tabela 11 - Descrição do UC 11

UC 11 – Executar Recomendação
Descrição sucinta: O usuário vendedor autoriza a venda do livro
Atores: Vendedor / Comprador
Pré-condições: Acessar sistema / possuir role Single / acessar pagina
Pós-condições: O usuário comprador recebe um email com os dados necessários para fechar o negocio
Fluxo básico: 1- Acessar o sistema 2- Realizar login com role de usuário vendedor 3- Acessar a pagina userS/relatórios/relRecoemndacoesLivrosVendedor.jsf 4- Executar a ação de recomendação 5- Fim do caso de uso
Fluxos alternativos: Inexistência de pessoas interesses nos livros
Fluxos de exceção: Não se aplica.
Estruturas de Dados: ED1, ED3, ED4 e ED5
Regras de negócios: RN4
Observações

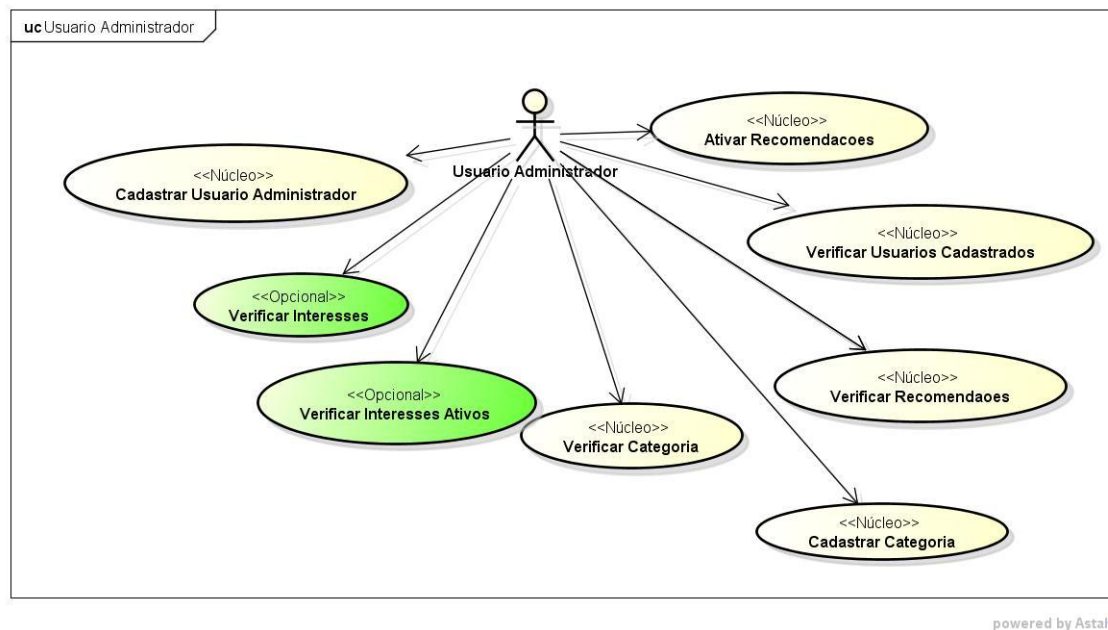


Figura 17 - Diagrama de Caso de Uso Administrador

Tabela 12 - Descrição do UC 12

UC 12 – Cadastrar Usuário Administrador
Descrição sucinta: Cadastrar um usuário com perfil de administrador no sistema.
Atores: Administrador
Pré-condições: Acessar sistema / possuir role Admin
Pós-condições: Mostrar a página usuarioAdministrador/cadastro/usuario.jsf
Fluxo básico:
1- Acessar o sistema
2- Realizar login com role de usuário administrador
3- Acessar a página Admin/cadastro/usuario.jsf
4- Preencher o formulário
5 Confirmar Cadastro
6 - Fim do caso de uso
Fluxos alternativos:
1 - O sistema exibe a mensagem de validação
2 - Retorna ao passo 3 do fluxo principal
Fluxos de exceção: Violação da RN6
Estruturas de Dados: ED1 e ED4
Regras de negócios:
Observações: Este caso de uso é semelhante ao cadastramento de usuário realizado pelo usuário público. Cabe a ressalva de que apenas este usuário pode criar um novo usuário com a role admin.

Tabela 13 - Descrição do UC 13

UC 13 – Verificar Interesses
Descrição sucinta: Verificação de todos os interesses cadastrados no sistema
Atores: Administrador
Pré-condições: Acessar sistema / possuir role Admin / existir interesses cadastrados
Pós-condições: Não se aplica
Fluxo básico:
1- Acessar o sistema
2- Realizar login com role de usuário administrador
3- Acessar a pagina Admin/relatórios/relInteresses.jsf
4- Verificar interesses cadastrados
5- Fim do caso de uso
Fluxos alternativos: Inexistência de interesses cadastrados
Fluxo de exceção - Não se aplica
Estruturas de dados: ED5 e ED6
Regras de negócio: RN12
Observações

Tabela 14 - Descrição do UC 14

UC 14 – Verificar Interesses ativos
Descrição sucinta: Verificação de todos os interesses ativos cadastrados.
Atores: Administrador
Pré-condições: Acessar sistema como Admin / acessar relInteresses.jsf
Pós-condições: Não se aplica
Fluxo básico:
1- Acessar o sistema
2- Realizar login com role de usuário administrador
3- Acessar a pagina Admin/relatórios/relInteresses.jsf
4- Verificar interesses ativos cadastrados
5- Fim do caso de uso
Fluxos alternativos: Inexistência de interesses cadastrados
Fluxo de exceção - Não se aplica
Estruturas de dados: ED5 e ED6
Regras de negócio: RN12
Observações

Tabela 15 - Descrição do UC 15

UC 15 – Verificar Categorias
Descrição sucinta: Verificar as categorias cadastradas no sistema
Atores: Administrador
Pré-condições: Acessar sistema como Admin / acessar relCategorias.jsf
Pós-condições: Não se aplica
Fluxo básico: 1- Acessar o sistema 2- Realizar login com role de usuário administrador 3- Acessar a página Admin/relatórios/relCategoria.jsf 4- Analisar as categorias cadastradas 5- Fim do caso de uso
Fluxos alternativos: Inexistência de interesses cadastrados
Fluxo de exceção - Não se aplica
Estruturas de dados: ED4
Regras de negócio: RN12
Observações

Tabela 16 - Descrição do UC 16

UC 16 – Cadastrar Categorias
Descrição sucinta: Realizar o cadastramento de uma nova categoria
Atores: Administrador
Pré-condições: Acessar como Admin / acessar pagina categoria.jsf
Pós-condições: Verificar cadastramento em Admin/relatorios/relCategoria.jsf
Fluxo básico: 1- Acessar o sistema 2- Realizar login com role de usuário administrador 3- Acessar a página Admin/cadastro/categoria.jsf 4- Preencher o formulário 5- Confirmar Cadastro 5- Fim do caso de uso
Fluxos alternativos: 1- O sistema exibe a mensagem de validação 2- Retorna ao passo 3 do fluxo principal
Fluxo de exceção - Violação da RN13
Estruturas de dados: ED1 e ED4
Regras de negócio:
Observações

Tabela 17 - Descrição do UC 17

UC 17 – Verificar usuário
Descrição sucinta: Verificar os usuários cadastrados no sistema
Atores: Administrador
Pré-condições: Acessar sistema como Admin / acessar relUsuarios.jsf
Pós-condições: Não se aplica
Fluxo básico: 1- Acessar o sistema 2- Realizar login com role de usuário administrador 3- Acessar a página Admin/relatórios/relUsuarios.jsf 4- Analisar os usuários cadastrados 5- Fim do caso de uso
Fluxos alternativos: Não se aplica
Fluxo de exceção - Não se aplica
Estruturas de dados: ED1 e ED2
Regras de negócio: RN1
Observações: Não se aplica.

Tabela 18 - Descrição do UC 18

UC 18 – Verificar Recomendações
Descrição sucinta: Verificar as recomendações cadastradas no sistema
Atores: Administrador
Pré-condições: Acessar sistema como Admin / Acessar relRecomendacoes.jsf
Pós-condições: Não se aplica
Fluxo básico: 1- Acessar o sistema 2- Realizar login com role de usuário administrador 3- Acessar a página Admin/relatórios/relRecomendacoes.jsf 4- Verificar recomendações cadastradas 5- Fim do caso de uso
Fluxos alternativos: Não se aplica
Fluxo de exceção - Não se aplica
Estruturas de dados: ED7 e ED8
Regras de negócio: RN11
Observações:

Tabela 19 - Descrição do UC 19

UC 19 – Ativar Recomendações
Descrição sucinta: Inicializar os agentes de software para geração das recomendações dos usuários compradores
Atores: Administrador
Pré-condições: Acessar o sistema / possui a role Admin / existir interesses e livros cadastrados com a categoria compatível com os interesses.
Pós-condições: Envio de email e Relatório a todos os interessados
Fluxo básico: 1- Acessar o sistema 2- Realizar login com role de usuário administrador 3- No menu do admin escolher a ação gerar recomendação 4- Acompanhar a instanciação dos agentes 5- Verificar as recomendações 6- Fim do caso de uso
Fluxos alternativos: Não se aplica
Fluxo de exceção - Não se aplica
Estruturas de dados: ED5, ED6, ED7 e ED8
Regras de negócio:
Observações:

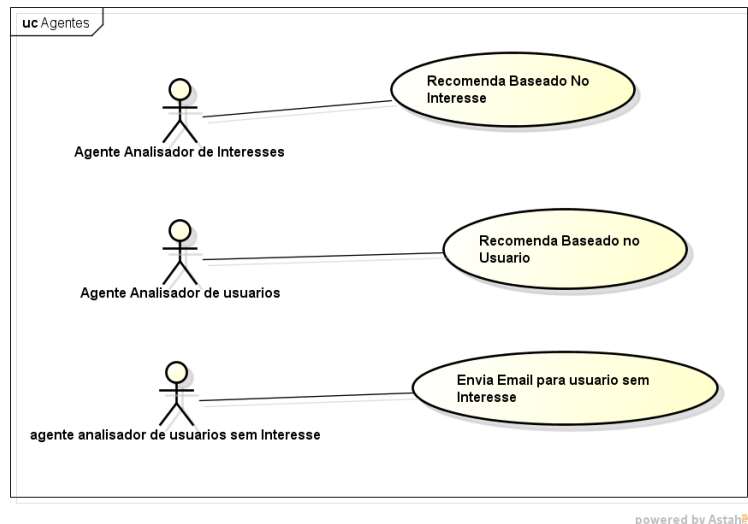


Figura 18 - Diagrama de Caso de Uso Agentes

Tabela 20 - Descrição do UC 20

UC 20 – Enviar email ao usuário comprador sem interesse

Descrição sucinta: Enviar email ao usuário comprador previamente cadastrados no sistemas que não possuem nenhum interesse associado.

Atores: Administrador/Agente analisador

Pré-condições: Acessar o sistema / possui a role Admin / existir pessoas sem interesses cadastrado.

Pós-condições: Envio de email e Relatório a todos os interessados

Fluxo básico:

- 1- Acessar o sistema
- 2- Realizar login com role de usuário administrador
- 3- No menu do admin escolher a ação gerar enviar email para pessoas SI
- 4- Acompanhar a instanciação dos agentes
- 5- Verificar o envio de email
- 6- Fim do caso de uso

Fluxos alternativos: Não se aplica

Fluxo de exceção - Não se aplica

Estruturas de dados: ED1e ED6

Regras de negócio: R7

Observações: Não se aplica.

Tabela 21 - Descrição do UC 21

UC 21 – Recomenda Baseado no Interesse
Descrição sucinta: Analisa os interesses ativos cadastrados no sistema que não possuem recomendação associada e envia email ao usuário comprador que cadastrou previamente este interesse.
Atores: Administrador/Agente recomendador
Pré-condições: Acessar o sistema / possuir a role Admin / existir interesses ativos cadastrados e sem nenhuma recomendação associada.
Pós-condições: Envio de email e Relatório a todos os interessados
Fluxo básico:
1- Acessar o sistema
2- Realizar login com role de usuário administrador
3- No menu do admin escolher a ação gerar recomendação baseado no interesse
4- Acompanhar a instanciação dos agentes
5- Verificar o envio de email
6- Fim do caso de uso
Fluxos alternativos: Não se aplica
Fluxo de exceção - Não se aplica
Estruturas de dados: ED1e ED6
Regras de negócio: R10, R11 e R12
Observações: Não se aplica.

Tabela 22 - Descrição do UC 22

UC 22 – Recomenda Baseado no Usuário
Descrição sucinta: Enviar email ao usuário comprador previamente cadastrados no sistemas que não possuem nenhum interesse associado.
Atores: Administrador / Agente recomendador
Pré-condições: Acessar o sistema / possuir a role Admin / existir pessoas com interesses ativos cadastrados sem nenhuma recomendação associada.
Pós-condições: Envio de email e Relatório a todos os interessados
Fluxo básico:
1- Acessar o sistema
2- Realizar login com role de usuário administrador
3- No menu do admin escolher a ação gerar recomendação baseado no usuário
4- Acompanhar a instanciação dos agentes
5- Verificar o envio de email
6- Fim do caso de uso
Fluxos alternativos: Não se aplica
Fluxo de exceção - Não se aplica
Estruturas de dados: ED1e ED6
Regras de negócio: R10, R11 e R12
Observações: Não se aplica.

3.3.2 Projeto do Domínio

Nesta seção verifica-se a arquitetura do *Free Recommendation System* sobre a perspectiva do projeto estrutural dos componentes do sistema, como o mesmo e cada um dos seus componentes serão organizados em termos de diretórios e como será empacotado para publicação. Neste sentido, a Figura 19 apresenta a ideia da organização dos arquivos de código do projeto. De interesse principal são as pastas *Java Resources* e *WebContent*. Na primeira pasta estão os arquivos da linguagem Java. Neles estão implementados os *controllers* e os *models* do padrão MVC.

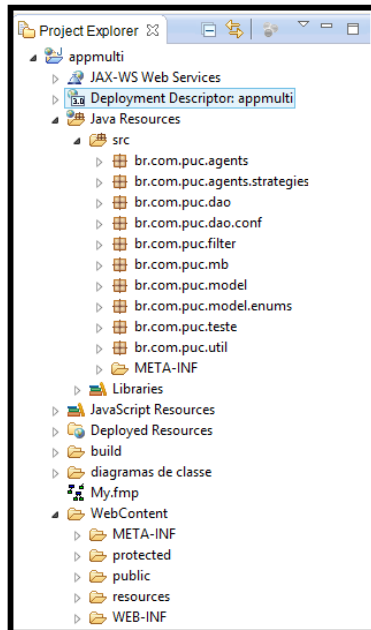


Figura 19 - Estrutura do Código Fonte

Ainda conforme a Figura 19, temos a pasta referente as páginas web que representarão a interface principal com os usuários do sistema e que compõem a camada *View* do padrão de projeto MVC.

3.3.2.1 Diagrama de Classes

3.3.2.1.1 Pacote DAO

Neste pacote estão implementados as classes responsáveis pela configuração e acesso a banco de dados através do DAO.

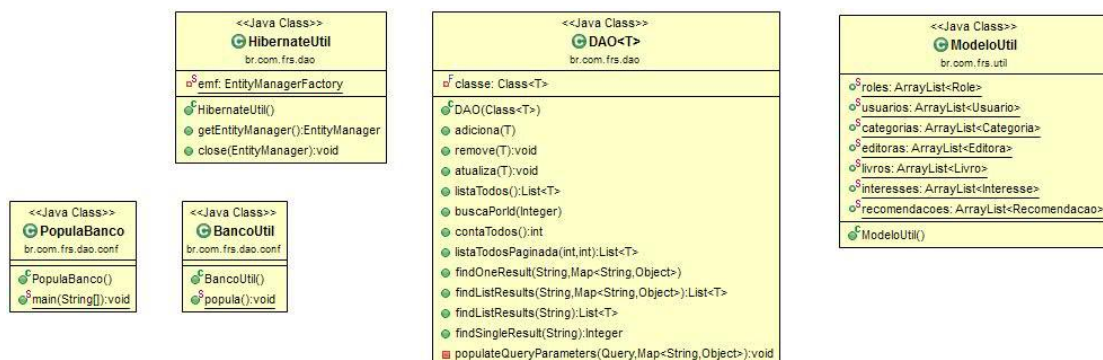


Figura 20 - Diagrama de Classes do Pacote DAO

3.3.2.1.2 Pacote Filter

Neste pacote estão implementados as classes essenciais para validação e autenticação dos usuários. Estas classes se encarregam de bloquear o acesso de páginas restritas de acordo com a role do usuário.

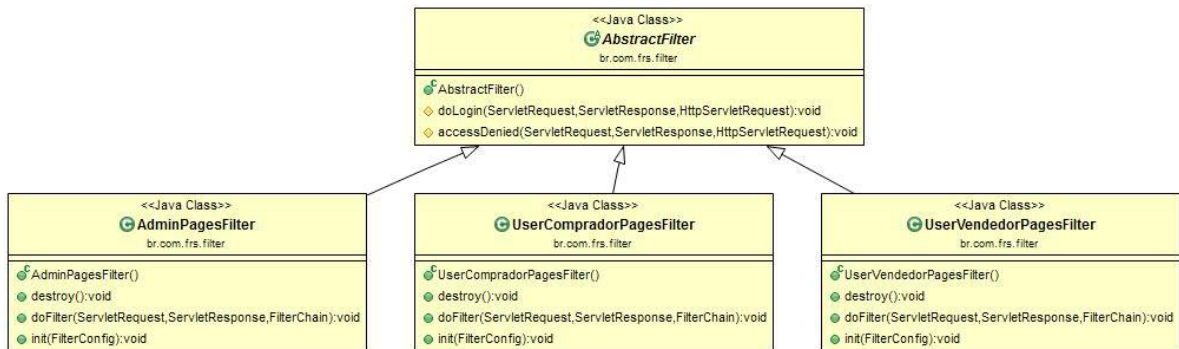


Figura 21 - Diagrama de Classes do Pacote Filter

3.3.2.1.3 Pacote MB

As classes do pacote *beans* atuam como uma fachada para a aplicação, além de deterem a maior parte das regras de negócio da aplicação.

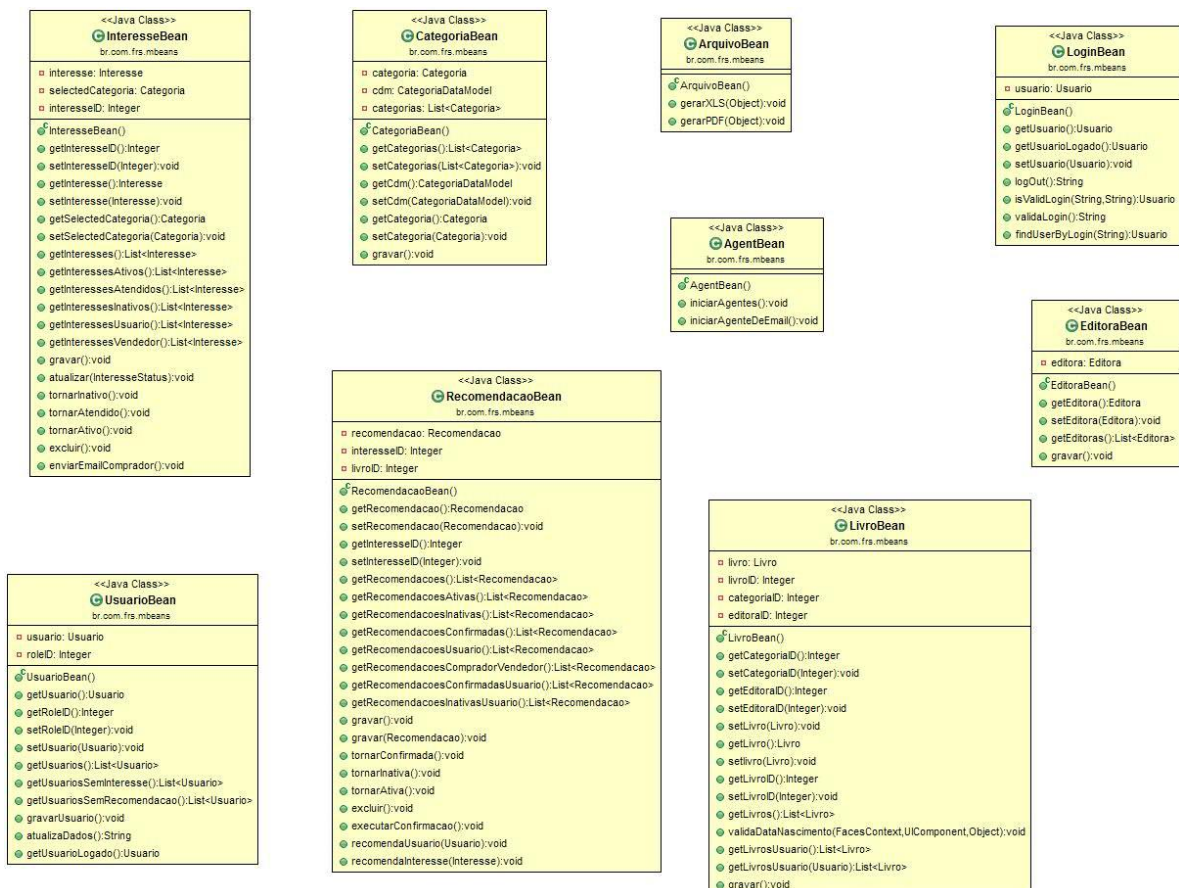


Figura 22 - Diagrama de Classes do Pacote MBean

3.3.2.1.4 Pacote Util

Neste pacote estão implementados as classes responsáveis por prover facilidades e funcionalidades genéricas que serão utilizadas pelo sistema.

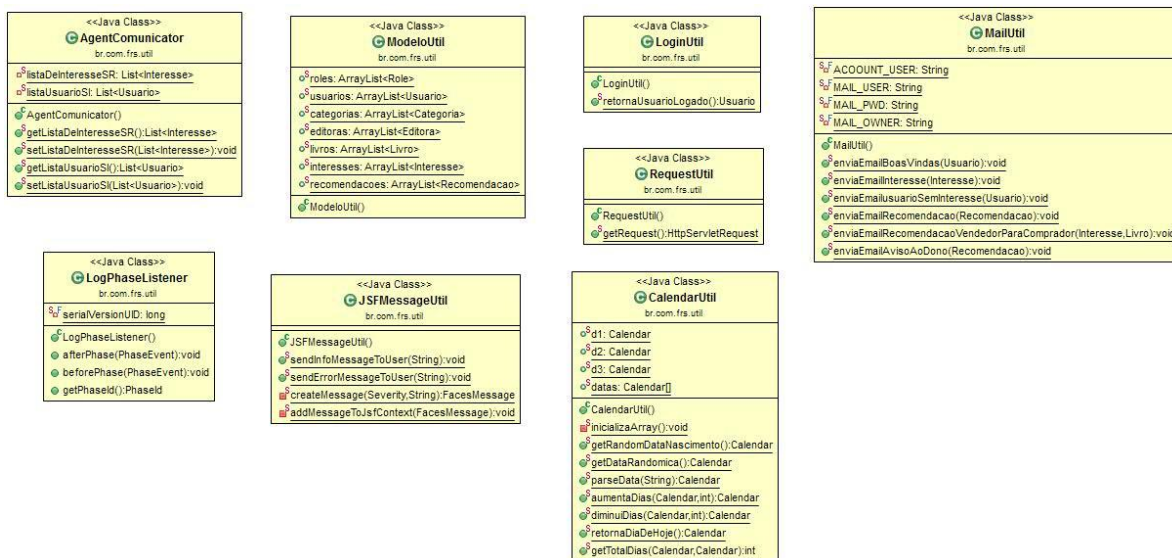


Figura 23 - Diagrama de Classes do Pacote Util

3.3.2.1.5 Pacote Model

Neste pacote são implementados as classes do modelo.

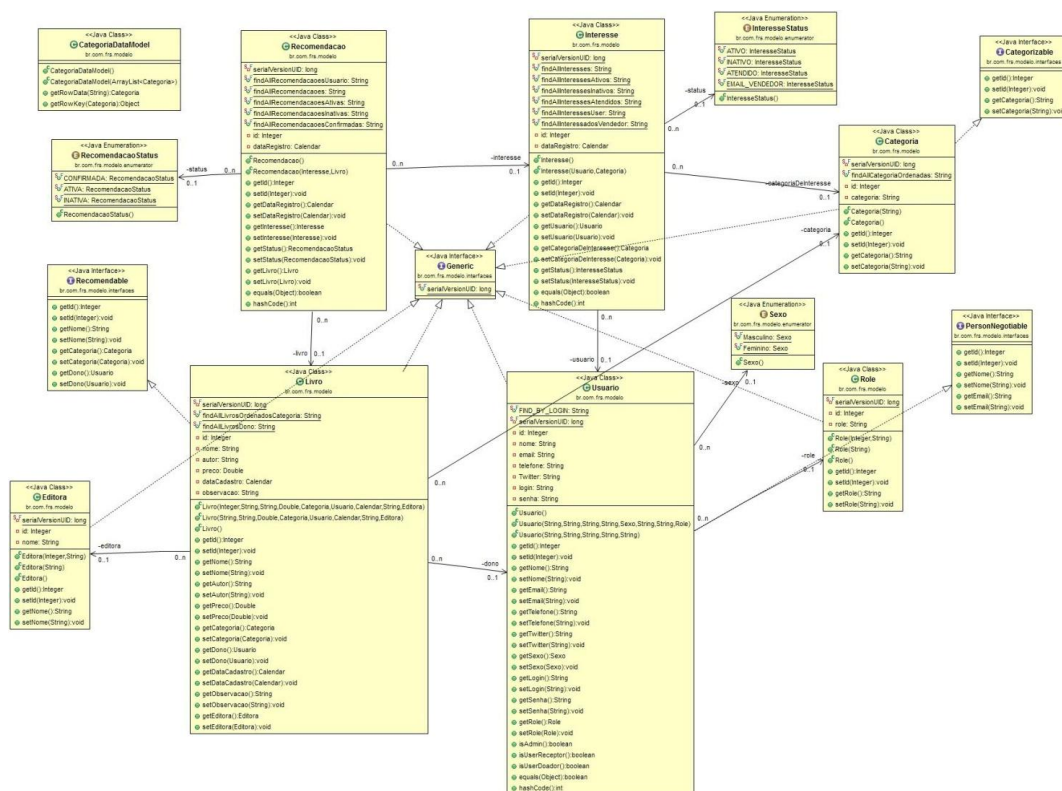


Figura 24 - Diagrama de Classe do Pacote Modelo

3.3.2.1.6 Pacote Agents

Neste pacote estão contidas todas as definições dos agentes do sistema bem como as definições de seus comportamentos.

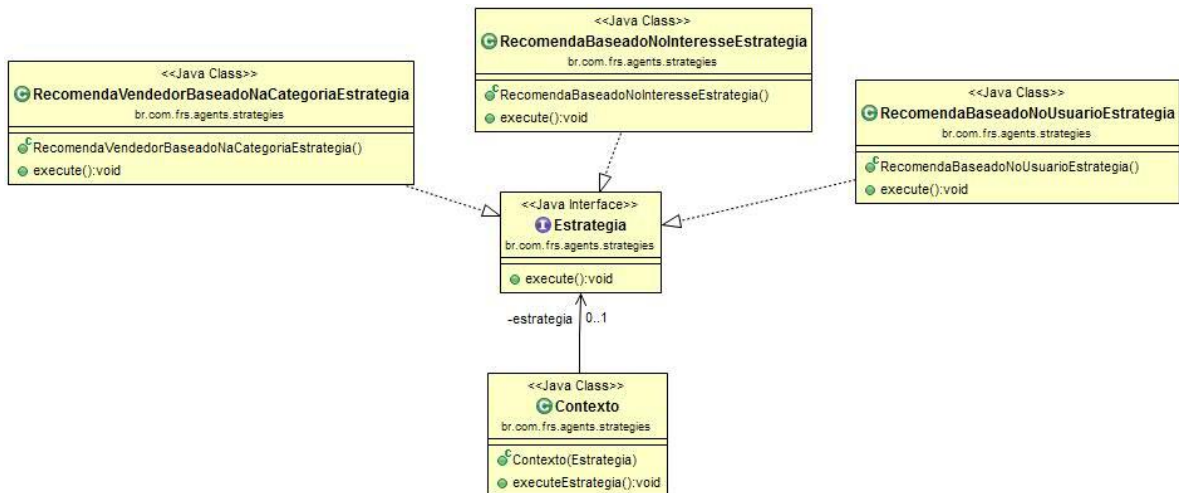


Figura 25 - Diagrama de Classe do Pacote Agents Strategies

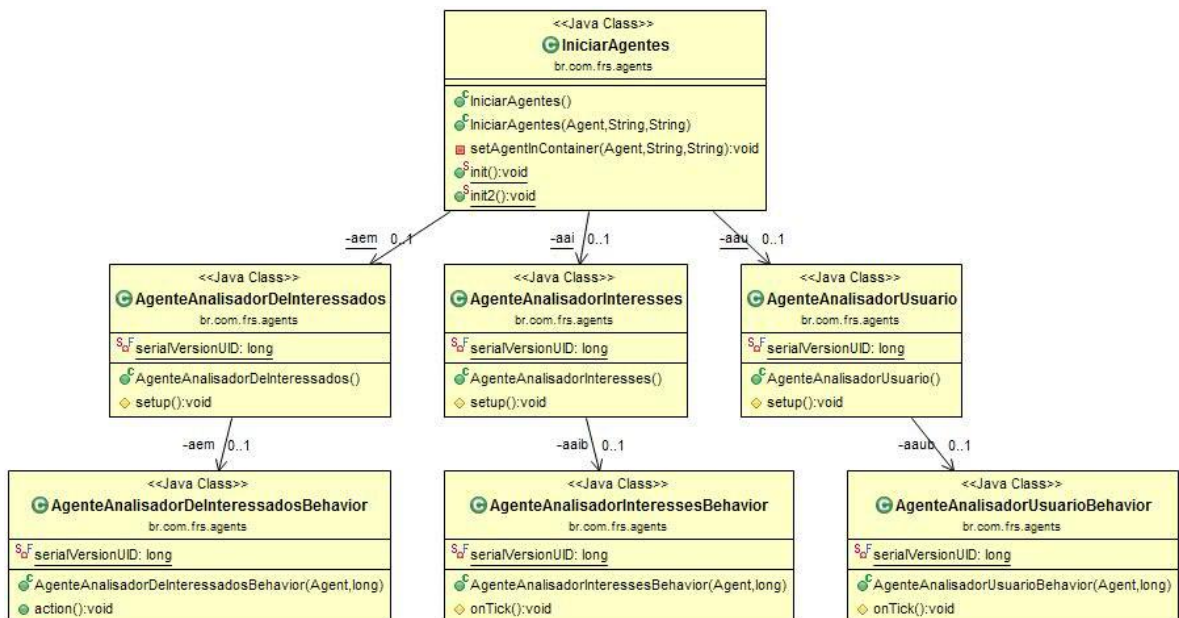


Figura 26 - Diagrama de Classe do Pacote Agents

3.3.3 Diagramas de Sequência

Alguns diagramas de sequência não foram realizados visto que o processo é o mesmo, só modificando os atores, neste sentido os mesmos foram desconsiderados neste documento.

3.3.3.1 Usuário Público

Para realização do cadastro no sistema o usuário acessa a página `usuario.jsf`. Nesta página ele terá que preencher um formulário com os dados de interesse para o sistema e após isso ele envia uma requisição que será controlada pela aplicação.

Caso os dados estejam válidos, um novo usuário será criado e persistido na base com a role de usuário escolhida (*comprador* ou *vendedor*).

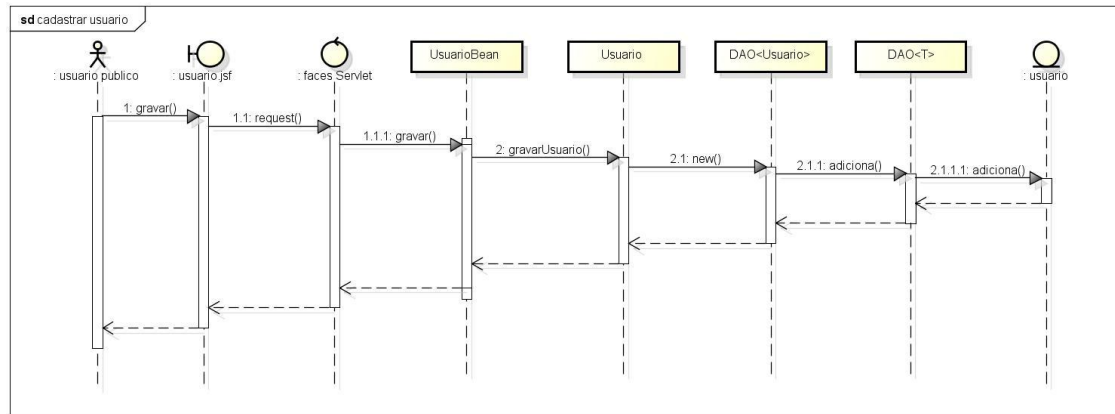


Figura 27 - Diagrama de Sequência Cadastrar Usuário

3.3.3.2 Usuário Comprador

Para realização do login o usuário terá que acessar a página login.jsf e preencher os dados do formulário (usuário e senha). Após isso é feita uma requisição para a aplicação que irá verificar na base de dados a existência deste usuário. Caso não exista será informado ao usuário o erro encontrado. Caso ele exista será verificado suas credenciais de acesso e o mesmo será redirecionado para a página relDadosUser.jsf em caso de sucesso.

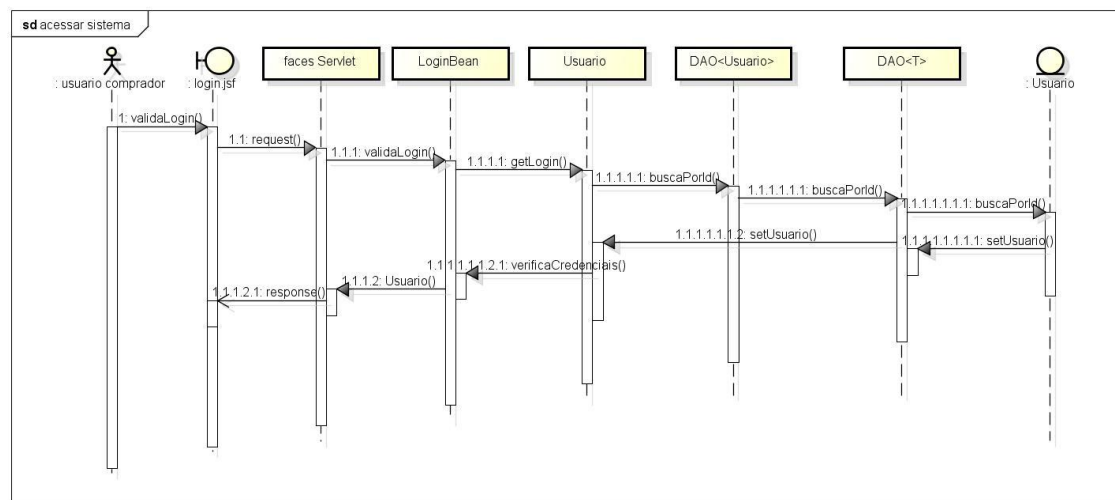


Figura 28 - Diagrama de Sequência Acessar Sistema

Após ter realizado o seu *Login*, o usuário comprador tem como página principal a relDadosUser.jsf e nesta pagina podem ser visualizados os dados cadastrados, de modo que o mesmo possa também alterá-los. Após a alteração dos dados, clica-se em salvar e os dados novos serão persistidos na base de dados.

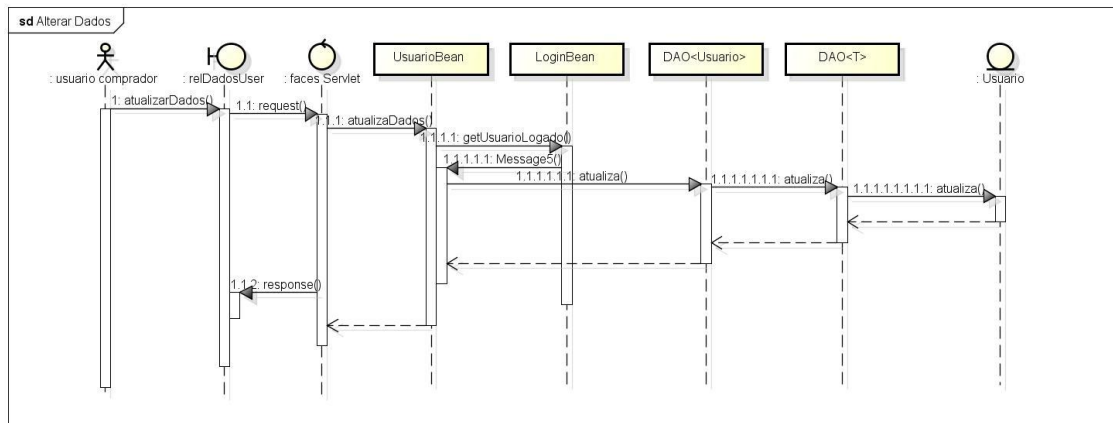


Figura 29 - Diagrama de Sequência Atualiza dados pessoais

Para cadastrar um interesse deverá estar logado no sistema. Em seguida deverá acessar a página interesses.jsf e inserir o interesse desejado. Feito isso é iniciado a comunicação entre o controller e o modelo visando persistir um novo interesse no banco e, após esse processo, é verificado através de mensagens o resultado (sucesso ou falha).

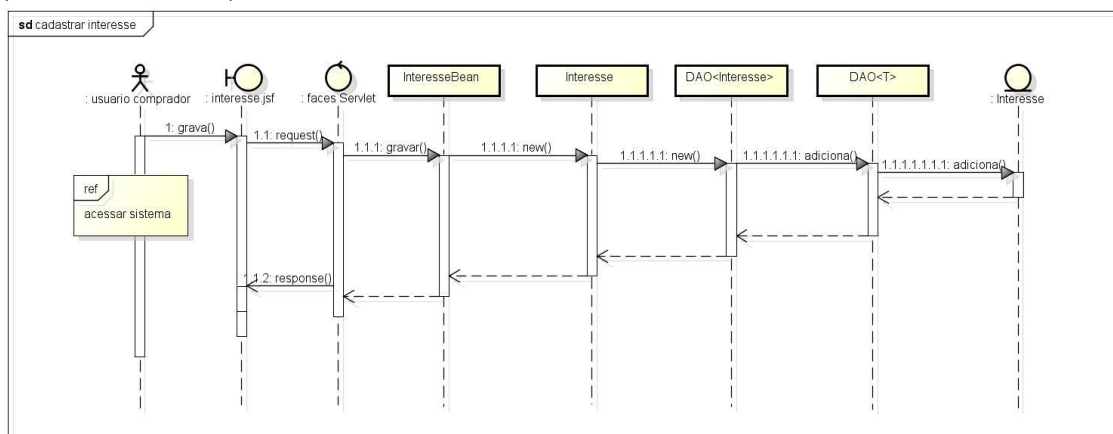


Figura 30 - Diagrama de Sequência Cadastra interesse

Após ter realizado o seu *Login*, o usuário comprador acessa a página relInteresses.jsf e nesta pagina podem ser visualizados os dados de interesses, de modo que o mesmo possa também alterá-los dentro das condições, ativo, inativo ou exclusão.

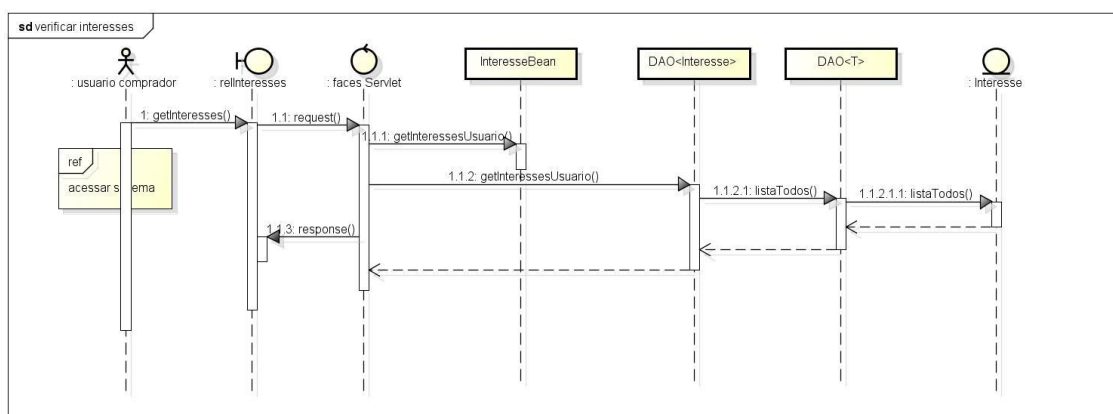


Figura 31 - Diagrama de Sequência Atualiza Status de Interesse

Para gerar relatório de interesses o usuário comprador deverá estar logado no sistema, uma vez que após a realização do *login* a variável usuário é colocada em uma sessão web acessar a página relInteresses.jsf e verificar todos os interesses cadastrados em sua conta.

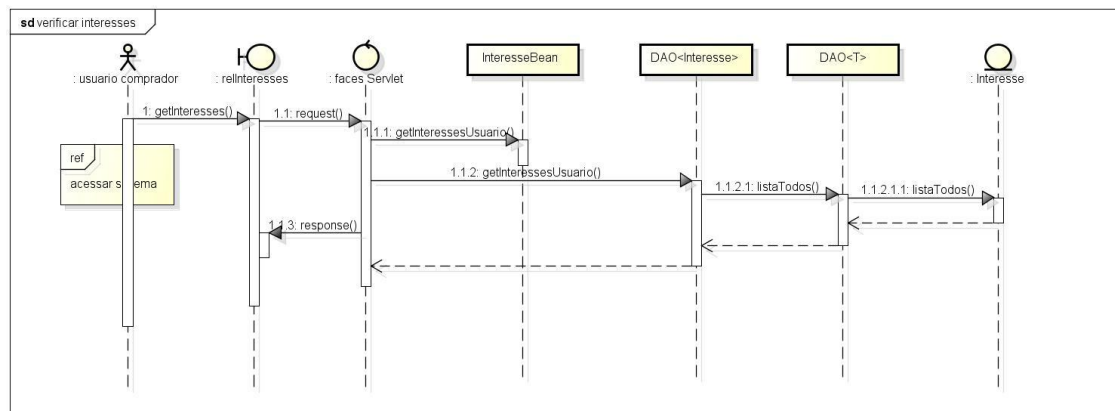


Figura 32 - Diagrama de Sequência Verificar Interesses

Para verificar as recomendações cadastradas no sistema o usuário comprador deverá estar logado no sistema, uma vez que após a realização do *login* a variável usuário é colocada em uma sessão web para monitoramento da interação de um determinado usuário com o sistema. Em seguida, acessar a página relRecomendacoesComprador.jsf. Feito isso é gerada uma requisição ao Faces Servlet que delega a resolução desta requisição ao InteresseBean. Este comunica-se com o modelo Interesse e com o DAO. Como retorno, uma lista de Recomendações relativas ao usuário é devolvida para a visão.

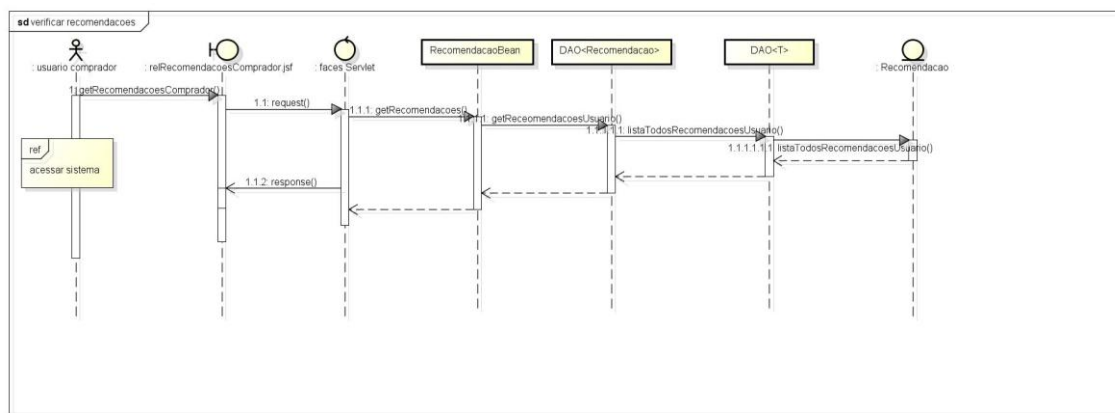


Figura 33 - Diagrama de Sequência Verificar Recomendações

Após ter realizado o seu *Login*, o usuário comprador acessa a página relRecomendacoes.jsf e nesta pagina podem ser visualizados os dados de interesses, de modo que o mesmo possa também alterá-los dentro das condições, ativo, inativo ou exclusão.

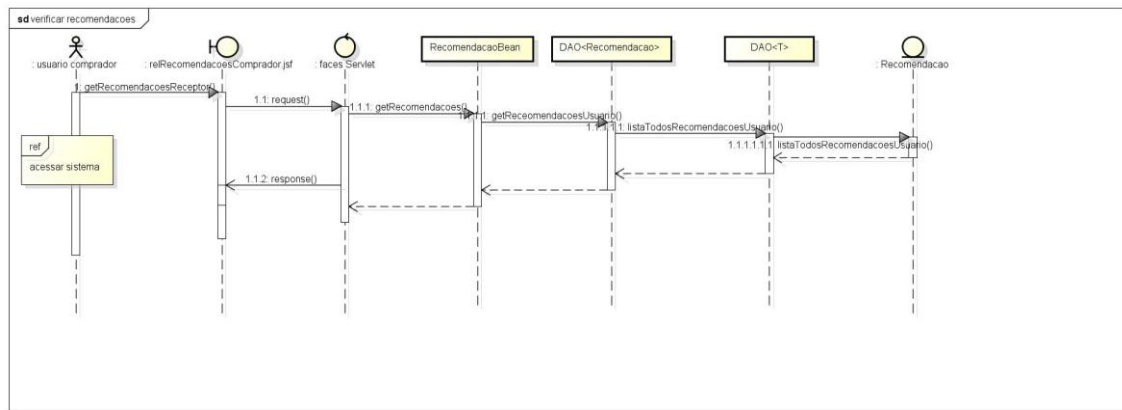


Figura 34 - Diagrama de Sequência Atualiza Status de Recomendações

3.3.3.3 Usuário Vendedor

Para cadastrar um livro o usuário vendedor deverá estar logado no sistema. Em seguida deverá acessar a página livro.jsf e inserir os dados relativos aos livros. Feito isso é iniciada a comunicação entre o *controller* e o modelo visando persistir um novo objeto do tipo livro no banco e, após esse processo, é verificado através de mensagens o resultado (sucesso ou falha).

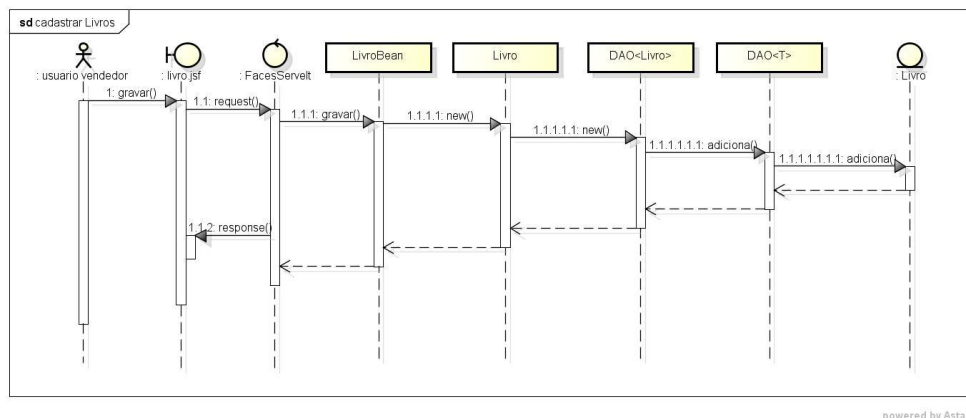


Figura 35 - Diagrama de Sequência Cadastrar Livros

Para verificar os livros cadastrados no sistema o usuário vendedor deverá estar logado no sistema, uma vez que após a realização do login a variável usuário é colocada em uma sessão web para monitoramento da interação de um determinado usuário com o sistema. Em seguida deverá acessar a página relLivros.jsf e verificar todos os livros que estão associados a sua conta.

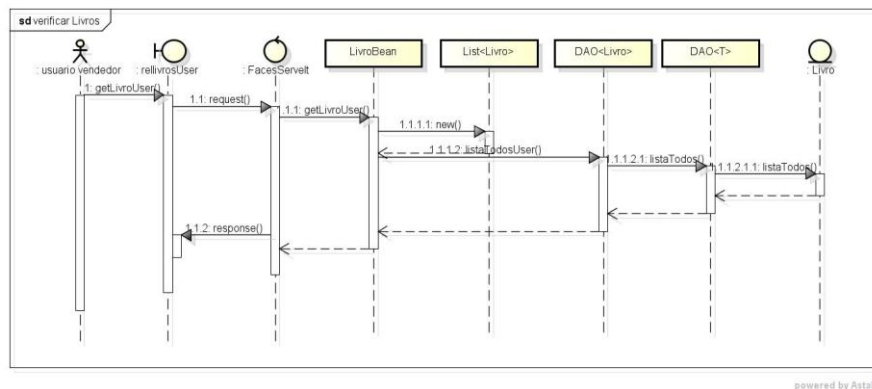


Figura 36 - Diagrama de Sequência Verificar Livros

Para verificar os interesses cadastrados no sistema o usuário vendedor deverá estar logado no sistema, uma vez que após a realização do login a variável usuário é colocada em uma sessão web para monitoramento da interação de um determinado usuário com o sistema. Em seguida deverá acessar a página relInteressadosVendedor.jsf e verificar todos os usuários compradores que tem interesse associado a algum livro do vendedor.

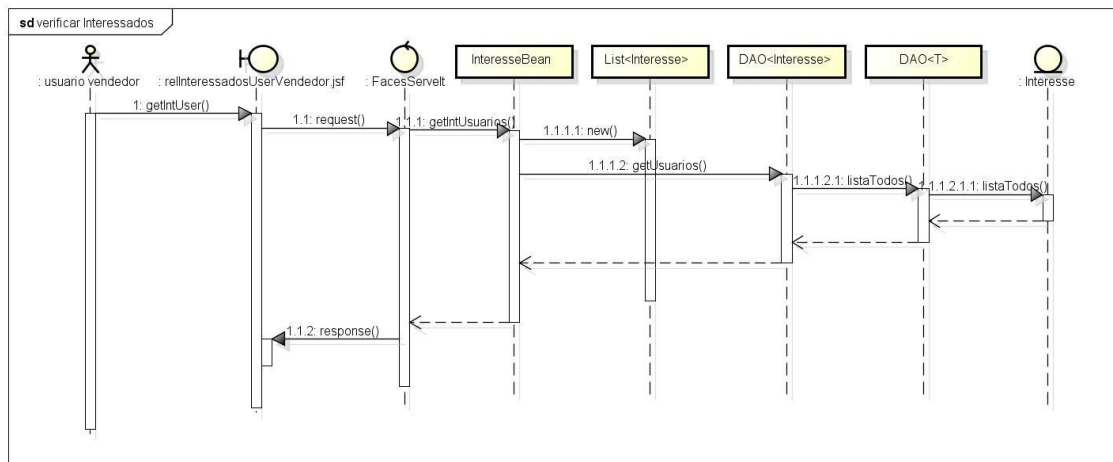


Figura 37 - Diagrama de Sequência Verificar Interessados

3.3.3.4 Usuário Administrador

Para realização o cadastro de uma nova categoria o usuário administrador terá que acessar a página login.jsf e preencher os dados do formulário (usuário e senha). Após isso, deverá acessar a página categoria.jsf e preencher um formulário informando os dados da nova categoria.

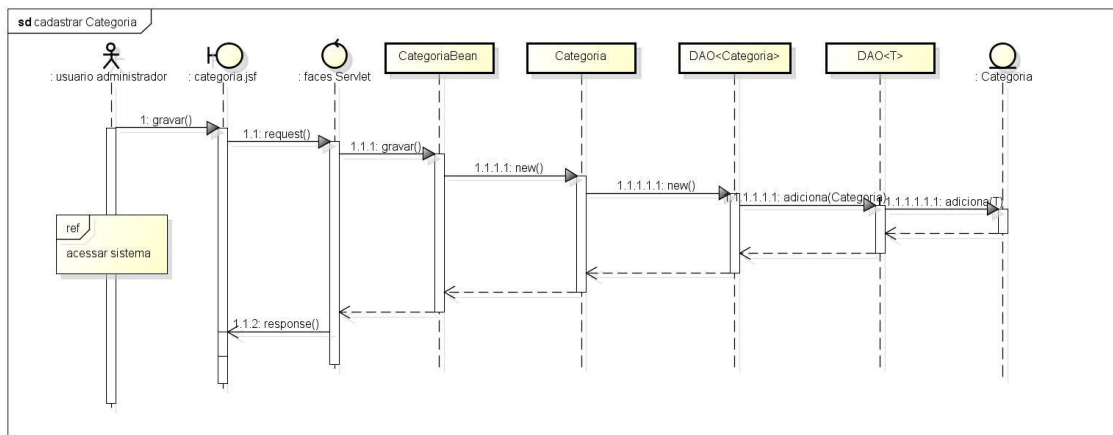


Figura 38 - Diagrama de Sequência Cadastrar Categoria

Para realizar o cadastro no sistema o usuário acessa a página usuário.jsf. Nesta página ele terá que preencher um formulário com os dados de usuário para o sistema e após isso, ele envia uma requisição que será controlada pela aplicação. Caso os dados estejam válidos, um novo usuário será criado e persistido na base com a role de usuário escolhida (Admin).

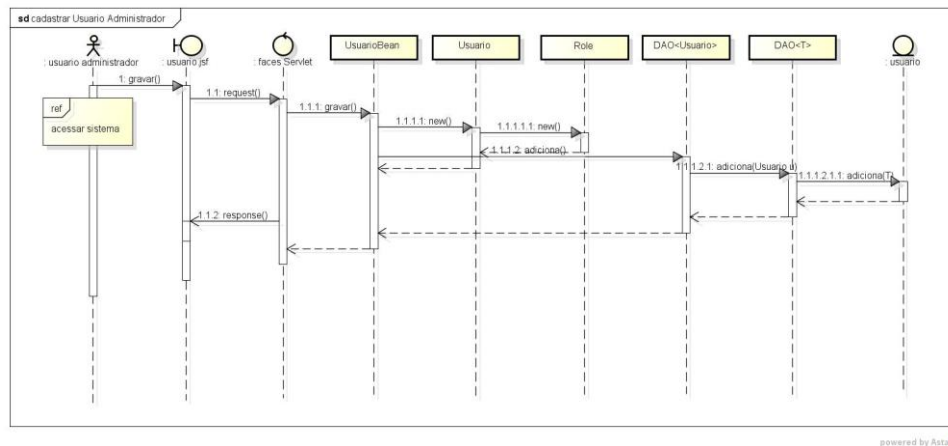


Figura 39 - Diagrama de Sequência Cadastrar Usuário Administrador

Para verificar as categorias cadastradas no sistema o usuário administrador terá que acessar a página login.jsf e preencher os dados do formulário (usuário e senha). Após isso, deverá acessar a página relCategoria.jsf. No momento do acesso a esta página é gerada uma requisição a aplicação que irá utilizar o CategoriaBean para verificar junto ao banco de dados quais são as categorias cadastrados. Como resposta a página é renderizada com uma lista de categorias que atendem aos requisitos elencados anteriormente.

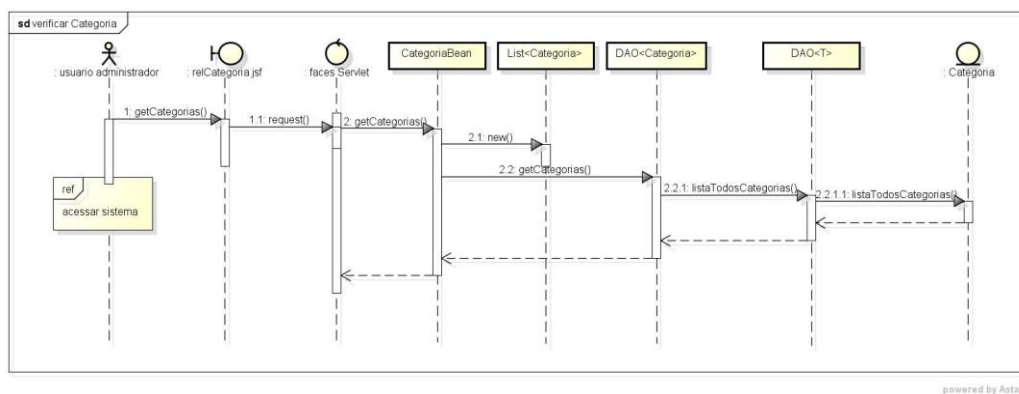


Figura 40 - Diagrama de Sequência Verificar Categorias

Para gerar relatório de recomendação o usuário administrador terá que acessar a página login.jsf e preencher os dados do formulário (usuário e senha). Após isso, deverá acessar a página relRecomendacoes.jsf. No momento do acesso a esta página é gerada uma requisição a aplicação que irá utilizar o RecomendacaoBean para verificar junto ao banco de dados quais são as recomendações independentemente do status associado.

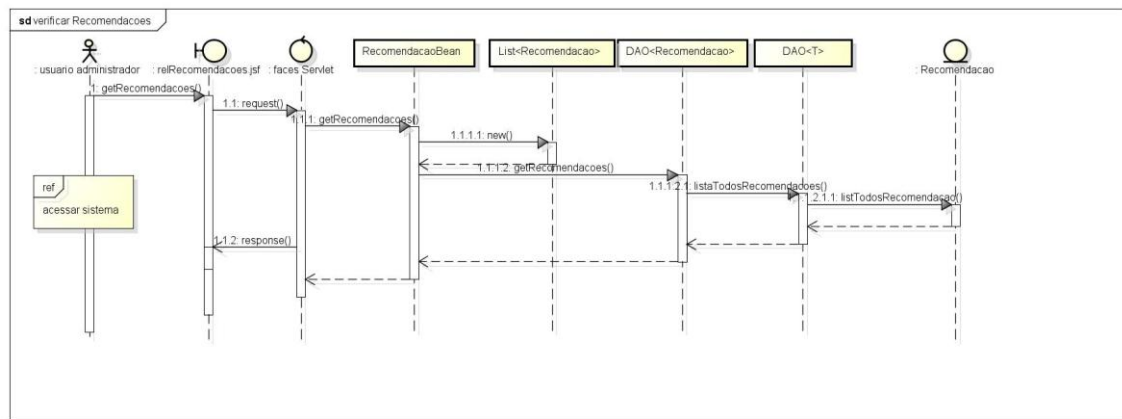


Figura 41 - Diagrama de Sequência Verificar Recomendação

3.3.3.5 Usuário Agente

Os usuários quando se cadastram no sistema não possuem necessariamente nenhum interesse associado. Caso este usuário continue sem nenhum interesse, ocasionalmente o administrador do sistema ativa a funcionalidade de enviar email para lembrar a estes usuários o cadastramento de interesses no sistema. Assim, o agente analisador de dados verifica através da classe AgentComunicator quem são os usuários que não possuem cadastro de interesse. Esta classe não acessa diretamente o Banco pois ela mantém uma lista estática de usuário com essas características. assim, o agente analisador extrai a informação do email do usuário e envia esta informação para a classe MailUtil. Esta por sua vez utiliza uma API para manipulação de email e envia uma mensagem para o usuário solicitando o cadastro de interesse no sistema.

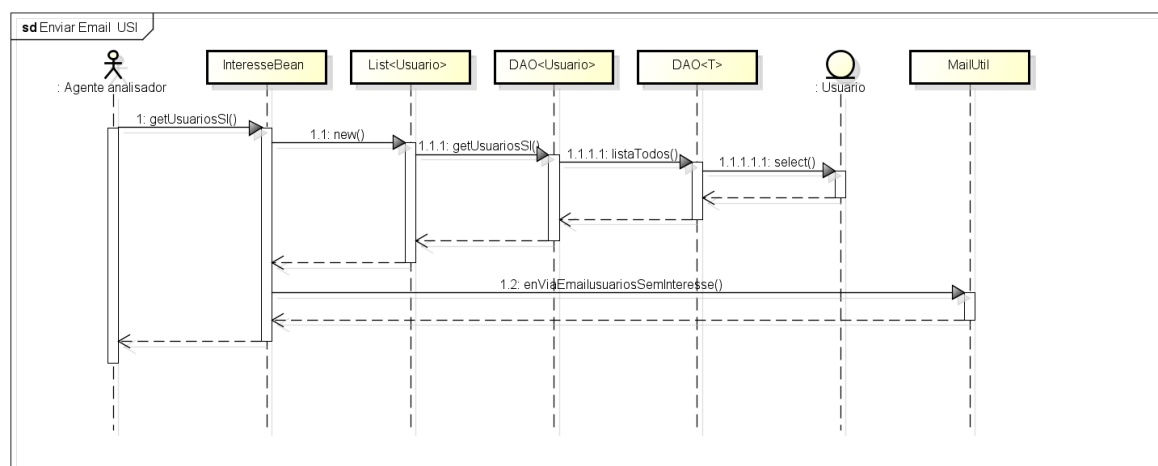
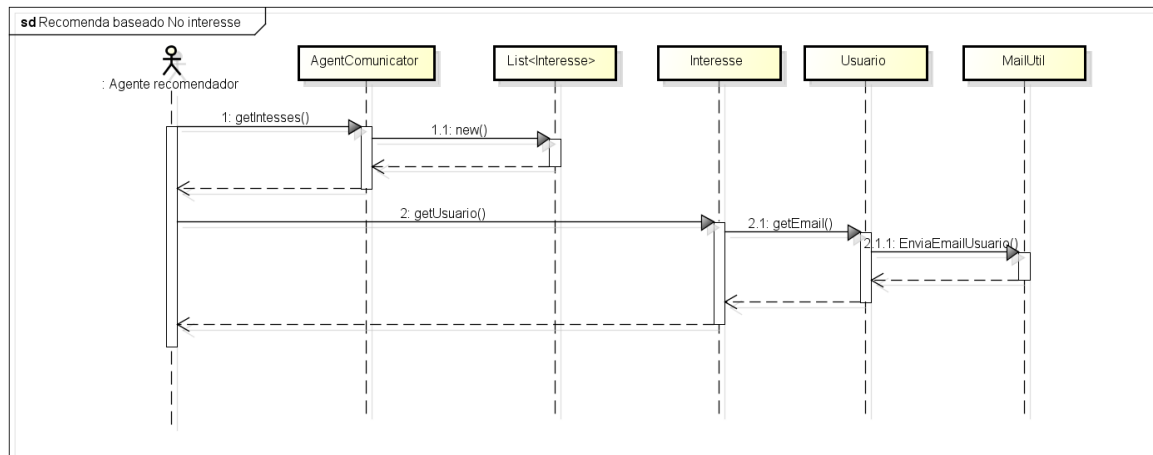


Figura 42 - Diagrama de Sequência Enviar Email usuários sem Interesse.

Os usuário quando se cadastram no sistema podem realizar o cadastramento de um ou mais interesses. Os interesses são baseados em categorias de Livros ou seja, dependendo do tipo de interesse, o usuário comprador pode ter um ou mais livros que se adéquem a esta categoria. Assim, o agente analisador de dados verifica através da classe AgentComunicator quem são os interesses ATIVOS que não possuem nenhuma recomendação associada. Esta classe não acessa diretamente o Banco pois ela mantém uma lista estática de Interesses com essas características.

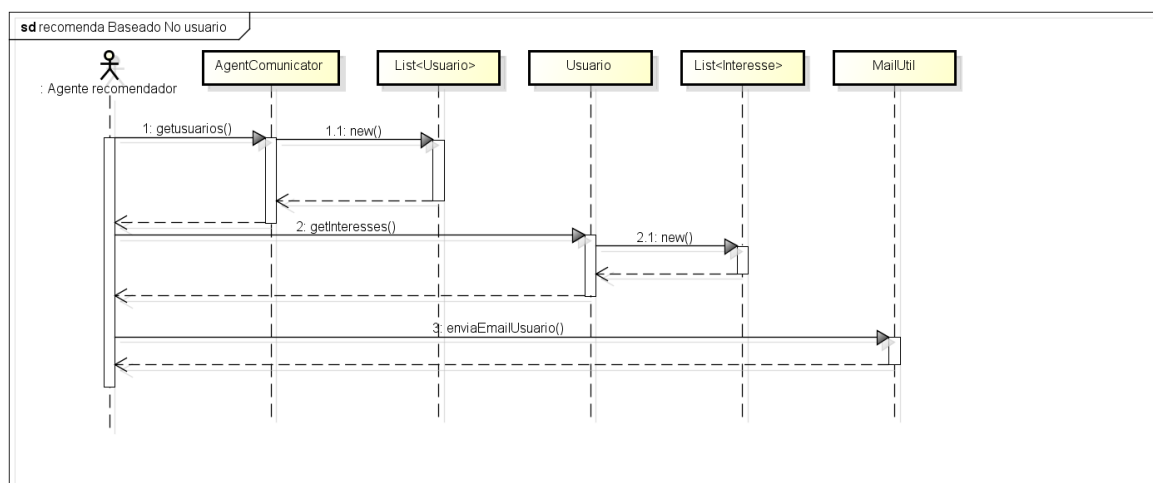
assim, o agente de análise de interesses extrai a informação do email do usuário e envia esta informação para a classe MailUtil. Esta por sua vez utiliza uma API para manipulação de email e envia uma mensagem para o usuário comprador com as características de um livro e os dados do proprietário do mesmo (usuário vendedor).



powered by Astah

Figura 43 - Diagrama de Sequência Recomenda baseado no Interesse.

Os usuário quando se cadastram no sistema podem realizar o cadastramento de um ou mais interesses. Os interesses são baseados em categorias de Livros ou seja, dependendo do tipo de interesse, o usuário comprador pode ter um ou mais livros que se adequam a esta categoria. Assim, o agente analisador de dados verifica através da classe AgentComunicator quem são os usuários que possuem um ou mais interesses ATIVOS que não possuem nenhuma recomendação associada. Esta classe não acessa diretamente o Banco pois ela mantém uma lista estática de Usuários com essas características. assim, o agente de análise de usuários extrai a informação do email do usuário e envia esta informação para a classe MailUtil. Esta por sua vez utiliza uma API para manipulação de email e envia uma mensagem para este usuário comcom as características de um livro e os dados do proprietário do mesmo (usuário vendedor).



powered by Astah

Figura 44 - Diagrama de Sequência Recomenda baseado no Usuário.

3.3.4 Projeto de Dados

Nesta seção será apresentada uma descrição da perspectiva de armazenamento de dados persistentes do sistema *FRS*. Neste sentido ela encontra-se organizada de acordo com os níveis de abstração de um banco de dados relacional típico, apresentando de modo distinto os modelos entidade-relacionamento, lógico e físico respectivamente.

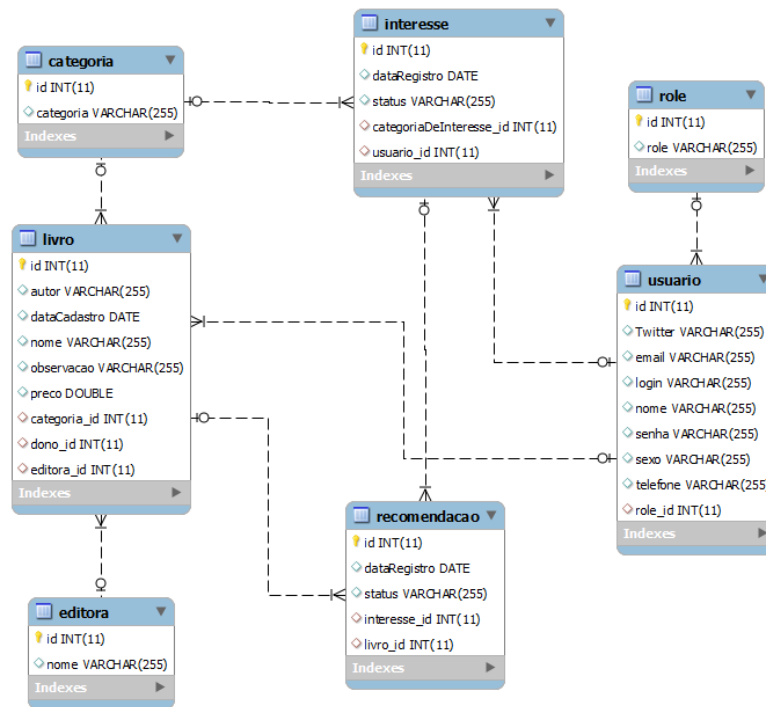


Figura 45 - Diagrama de Lógico de dados

4 Guia de Instalação e Configuração

Na pasta raiz da mídia em anexo a este projeto podemos verificar a existência de uma árvore de diretórios descrita na **Figura 37**. Para instalação dos itens a seguir pode ser utilizado um computador com qualquer plataforma de sistema operacional.

4.1 Instalação no servidor web

Na pasta instalação -> arquivo.war encontra-se um arquivo chamado frs.war. Para configuração deste projeto no servidor web Tomcat devem ser feitos os seguintes passos:

- 1) Instale o MySQL/MySQL Workbench no computador
- 2) Crie um usuário com as seguintes configurações: nome: root / senha: root
- 3) Configure suas permissões para select, insert, delete, update
- 4) Crie um banco chamado frsdb.

5) Para carregar o banco, temos duas opções:

5.1) Abra o programa MySQL WorkBench e na aba SQLEditor coloque o conteúdo do arquivo frsdb.sql que encontra-se no diretório instalação -> Banco de Dados -> frsdb.sql.

5.2) Em seguida clique em Execute SQL (Botão de raio).

6) Caso não possua o Tomcat instalado em sua máquina. Faça o download deste servidor web na versão 7.0.

7) Para realizar o deployment Copie e cole o arquivo frs.war (Instalação -> Arquivo .war) na pasta PATH/webapps do seu Tomcat, onde PATH é o caminho local do seu Tomcat.

8) Reinicie seu Tomcat.

9) Acesse http://localhost:8080/frs_puc/ e clique em instruções para verificar as configurações de usuários.

4.1.1 Instalação na IDE

Na pasta código-fonte encontra-se um arquivo chamado frs.zip. Para configuração deste projeto na IDE Eclipse devem ser feitos os seguintes passos:

1) Caso seu eclipse não possua uma pasta para workspace, crie uma pasta (por exemplo workspacePSS) e salve o arquivo frs.zip na mesma.

2) No momento em que for executado o eclipse, configurar o workspace para a pasta criada/ou já configurada no passo anterior.

3) Clique em Import -> General ->Existing Project into workspace

4) Selecione a opção Select archive file.

5) Em seguida clique em Browse e aponte para o arquivo frs.zip que encontra-se no workspace.

6) Clique em Next

7) Importe todos os arquivos

8) Ao final deste passo, clique com o botão direito em cima do projeto na aba Projects e selecione Build Path -> Configure Build Path

9) Verifique as dependências do projeto e corrija se for necessário.

5 Anexo

5.1 Estruturas de Dados

Estrutura	Descrição
ED1	Classe Usuário
ED2	Classe Role
ED3	Classe Livros
ED4	Classe Categoria
ED5	Classe Interesses
ED6	Classe Enumeration Interesse Status
ED7	Classe An[alise de
ED8	Classe Enumeration An[alise de Status

5.2 Regras de Negócio

Regra	Descrição
R1	<p>Durante o preenchimento do formulário um usuário público deve preencher os dados da ED1, contudo a ED2 já será preenchida automaticamente pelo sistema com a Role="User".</p> <p>Todos os campos devem ser preenchidos onde as regras de validação encontram-se descritas no arquivo usuário.xhtml</p> <p>Após o preenchimento o usuário é redirecionado para a pagina login.jsf</p>
R2	<p>O usuário para acessar o sistema deve possuir um cadastro no sistema. Para efetuar tal cadastro observe RN1.</p> <p>Os usuários sem cadastro poderão ver apenas as páginas no pacote público. Os usuário com cadastro poderão ver as páginas no pacote protected de acordo com sua role (Admin, usuarioVendedor ou usuario-Comprador). Para respeitar essas regras foram implementados filtros (veja em br.com.frs.filter)</p> <p>Os dados de Login e senha deverão ser preenchidos obrigatoriamente seguindo as regras de validação (login.jsf)</p>
R3	<p>Para realizar o cadastro do livro o usuário vendedor deverá possuir credenciais para acesso no sistema. O usuário pode fazer cadastro de diversos livros de acordo com os dados contidos em ED2, ED3 e ED4. Existe uma série de validações no formulário e estas estão descritas em livro.jsf</p>
R4	<p>Com o usuário logado temos a variável usuário (ED1) na sessão e temos condições de identificá-lo.</p>
R5	<p>Todos os campos devem ser preenchidos onde as regras de validação encontram-se descritas no arquivo usuario.jsf</p>
R6	<p>Ao sair do sistema, os dados da sessão web serão finalizados e não poderão ser recuperados sem a realização de um novo login.</p>

R7	O cadastramento de interesse é de extrema importância para o funcionamento do sistema. Para cadastrar um interesse, este deverá ser associado a um usuário. Poderá ser cadastrados diversos interesses para cada usuário. Caso o usuário permaneça sem nenhum interesse associado um agente ficará encarregado de alertar este usuário para o cadastramento de interesses.
R8	Todos os campos devem ser preenchidos onde as regras de validação encontram-se descritas no arquivo interesse.jsf
R9	Os interesses possuem status: ATIVO – Significa que o interesse foi cadastrado com sucesso e aguarda confirmação. INATIVO – Significa que o usuário desistiu (por algum motivo) deste interesse. ATENDIDO – Significa que este interesse possui alguma recomendação associada.
R10	As recomendações serão baseadas nos interesses previamente cadastrados. Para cada interesse, várias recomendações podem ser realizadas. Caso algum interesse permaneça sem recomendação, um agente ficará encarregado de realizar a recomendação de livros compatível com a categoria de interesse do usuário.
R11	As recomendações possuem os seguintes status: ATIVA – é uma recomendação recém realizada que não foi confirmada pelo comprador e vendedor; INATIVA- é uma recomendação que não foi concretizada pelo vendedor/comprador por algum motivo ;CONFIRMADA – é uma recomendação que foi concretizada pelo vendedor/comprador por algum motivo.
R12	As recomendações serão realizadas por agentes de software. Estes podem realizar análise de com base no interesse ou com base no usuário (este possui uma serie de interesses). Para cada análise apenas um interesse será atendido e um livro relacionado. Uma pessoa pode ter várias recomendações para o mesmo interesse, desde que este esteja com status não confirmado.