

UNIVERSIDADE FEDERAL DE JUIZ DE FORA
PÓS-GRADUAÇÃO EM DESENVOLVIMENTO COM TECNOLOGIA JAVA

PAULO CESAR DE SOUZA PINHEIRO

SISIMÓVEIS
SISTEMA DE AGENDAMENTO DE IMÓVEIS

Juiz de Fora

2013

PAULO CESAR DE SOUZA PINHEIRO

SISIMÓVEIS
SISTEMA DE AGENDAMENTO DE IMÓVEIS

Trabalho de Conclusão de Curso
apresentado como requisito para
obtenção do título da Pós-Graduação
em desenvolvimento com tecnologia
Java da Universidade Federal de Juiz
de Fora.

ORIENTADOR: TARCISIO DE SOUZA LIMA

Juiz de Fora
2013

PAULO CESAR DE SOUZA PINHEIRO

SISIMÓVEIS
SISTEMA DE AGENDAMENTO DE IMÓVEIS

Trabalho de Conclusão de Curso apresentado como requisito para obtenção do título de Pós-Graduação em desenvolvimento com tecnologia Java da Universidade Federal de Juiz de Fora.

Juiz de Fora, 30 de Junho de 2013

Prof. MSc Tarcisio de Souza Lima - Orientador
Universidade Federal de Juiz de Fora

Prof. DSc Michel Heluey Fortuna - Examinador
Universidade Federal de Juiz de Fora

Prof. MSc Marcos Henrique Fonseca Ribeiro - Examinador
Universidade Federal de Juiz de Fora

Juiz de Fora
2013

AGRADECIMENTOS

Agradeço primeiramente a Deus e aos meus pais pela existência, a minha esposa pelo apoio, aos amigos pelo companheirismo, Aos professores da pós-graduação pelo desenvolvimento pessoal e profissional e ao professor Tarcisio por aceitar me orientar. A todos o meu sincero muito obrigado.

“As boas obras não tornam bom o homem, mas o homem bom pratica boas obras. As más obras não tornam mau o homem, mas o homem mau pratica obras más”

(Martinho Lutero)

RESUMO

Este trabalho apresenta como proposta o desenvolvimento de um sistema de informação para controlar os agendamentos de Imóveis. Atualmente, as empresas buscam o apoio de tais sistemas para viabilizar os agendamentos de imóveis para agilizar estes serviços e oferecer melhorias na prestação dos serviços.

No desenvolvimento proposto neste trabalho será aplicada a metodologia RUP com o padrão de notação UML, o padrão de projeto MVC, bem como o conhecimento adquirido durante o curso. Desta forma iremos oferecer um sistema Web, com acessibilidade de múltiplos usuários e que será construído em Java.

Palavras-Chave: RUP, UML, Padrões de projeto, MVC, Web, controle de agendamento de imóveis.

LISTA DE ILUSTRAÇÕES

Ilustração 1 - Estrutura do processo, relacionamento entre as fases e o esforço de desenvolvimento em cada fluxo de trabalho de processo - (KRUTCHEN, 2003). . .	13
Ilustração 4.1 - Diagrama de Casos de uso do SISIMÓVEIS	21
Ilustração 5.1 - Diagrama de Classes de Entidade.	26
Ilustração 5.3 - Diagrama de Sequência Incluir Imóveis	28
Ilustração 5.3.2 - Diagrama de Sequência Excluir Imóveis	30
Ilustração 5.3.3 - Diagrama de Sequência Alterar Imóveis	32
Ilustração 5.3.4 - Diagrama de Sequência Consultar Imóveis	34
Ilustração 5.4.2 - MVC Model 2 pela Sun (2004)	36
Ilustração 5.4.3 - Arquitetura em camadas.	38
Ilustração 6.1 – Divisão do SISIMÓVEIS.	40
Ilustração 6.2 – Divisão das classes DAO e CONTROLLER	41
Ilustração 6.2.1 – Mapeamento da classe Imóvel.	44
Ilustração 6.3.1 – Tela principal do Sismóveis	45
Ilustração 6.3.2 – Tela manter Imóvel	46
Ilustração 6.3.3 – Tela manutenção da Imóvel	47

LISTA DE SIGLAS

RUP: Rational Unified Process

UML: Unified Modeling Language

MVC: Model-view-controller

SISIMÓVEIS: Sistema de Agendamento de Imóveis

SUMÁRIO

1 INTRODUÇÃO	11
1.1 Objetivos gerais.....	11
1.2 Metodologia aplicada.....	11
2 MODELAGEM COMERCIAL	15
2.1 Introdução	15
2.2 Descrição geral	17
2.3 Funções do produto	17
3 REQUISITOS ESPECIFICOS	19
3.1 Requisitos funcionais.....	19
3.2 Requisitos não funcionais.....	20
4 FLUXO DE REQUISITOS	20
4.1 Diagrama de caso de uso	21
4.2 Caso de uso Manter Imóvel	22
5 ANÁLISE E PROJETO	25
5.1 - Diagrama de classes.....	26
5.2 - Especificação do Caso de Uso Manter Imóveis	27
5.3 - Diagramas de Sequência	28
5.3.1 - Diagrama de Sequência Incluir Imóveis	28
5.3.2 - Diagrama de Sequência Excluir Imóveis	30
5.3.3 - Diagrama de Sequência Alterar Imóveis	32
5.3.4 - Diagrama de Sequência Consultar Imóveis	34
5.4.1 - Padrões de projeto	36
5.4.2 – Abordagem MVC	36
5.4.3 - Arquitetura em camadas	37

6 IMPLEMENTAÇÃO	40
6.1 – Visão geral	40
6.2 – Aplicação das tecnologias	43
6.2.1 – JDBC	43
6.3 – Apresentação da aplicação.....	45
7 CONSIDERAÇÕES FINAIS.	48
Referências	49
Apêndices	50
Apêndice A – Demais Casos de Uso do SISIMÒVEIS.....	50
Diagrama de Tabelas Relacionais (DTR).....	89
Apêndice B – Principais telas do SISIMÒVEIS.....	90

1 INTRODUÇÃO

1.1 Objetivos gerais

Buscando aplicar alguns conhecimentos que foram mostrados na pós-graduação de desenvolvimento de sistemas com tecnologia Java, juntamente com meu orientador Tarcisio de Souza Lima, modelamos e codificamos o SisImóveis, sistema de agendamento de imóveis, procurando demonstrar de forma teórica e prática o desenvolvimento deste trabalho, para obtenção do título deste curso. No apêndice B serão encontradas algumas dessas telas, que serão exibidas para maiores esclarecimentos.

Para alcançar este objetivo deve-se utilizar o processo de engenharia de software RUP como base para o desenvolvimento da aplicação. O emprego desta metodologia visa atingir alta qualidade no software, satisfazendo as necessidades da empresa. (KRUCHTEN, 2003).

1.2 Metodologia aplicada.

Neste trabalho será usado o processo de engenharia de software RUP, seguindo o paradigma da Orientação a Objetos e fazendo uso do padrão de notação proposta pela UML, sendo esta a metodologia aplicada.

A UML é uma linguagem padrão para modelagem de sistemas com a finalidade de auxiliar no desenvolvimento dos mesmos, ela inclui técnicas de notações gráficas largamente utilizadas, como o Diagrama de Classes, Diagrama de Casos de Uso e Diagramas de Sequência, que serão explicados e mostrados posteriormente. (FOWLER, 2005)

Tal padrão será usado neste trabalho para visualização, especificação e construção do software.

Para o desenvolvimento deste sistema será empregado o RUP que é um processo de engenharia de software fornecido pela Rational Software. Ele fornece um modelo de trabalho disciplinado, apontando tarefas e responsabilidades dentro de um grupo de trabalho. Seu objetivo é assegurar o desenvolvimento de software de alta qualidade, atendendo os requerimentos de seus usuários com orçamento e prazo realistas.

O RUP é um guia de como usar o padrão UML de forma eficaz, melhorando a produtividade da equipe de desenvolvimento, sendo altamente configurável, o que permite que organizações de todo o tamanho o usem. Tal processo é apoiado por ferramentas, que auxiliam em grandes partes do trabalho, estas sendo usadas para criar e manter várias etapas do desenvolvimento, como: modelagem visual, programação, teste e etc.

O *Rational Unified Process* descreve como eficazmente usar técnicas comercialmente aprovadas de desenvolvimento, que são as “melhores práticas de engenharia de software” e são chamadas assim pois são fruto da observação do que é usado em organizações de sucesso. Tais práticas são: (KRUCHTEN, 2003).

- controlar mudanças no software.
- verificar continuamente a qualidade de software;
- gerenciar exigências;
- desenvolver software interativamente;
- usar arquiteturas baseadas em componente;
- modelar visualmente o software;

O processo pode ser demonstrado em duas dimensões: o eixo horizontal representa o tempo, dinâmico, e mostra os aspectos do ciclo de vida, ou seja, as fases do processo e como este se desdobra. O eixo vertical representa o aspecto estático do processo, atividades e fluxos de trabalho, mostrados na Ilustração 1 a seguir.

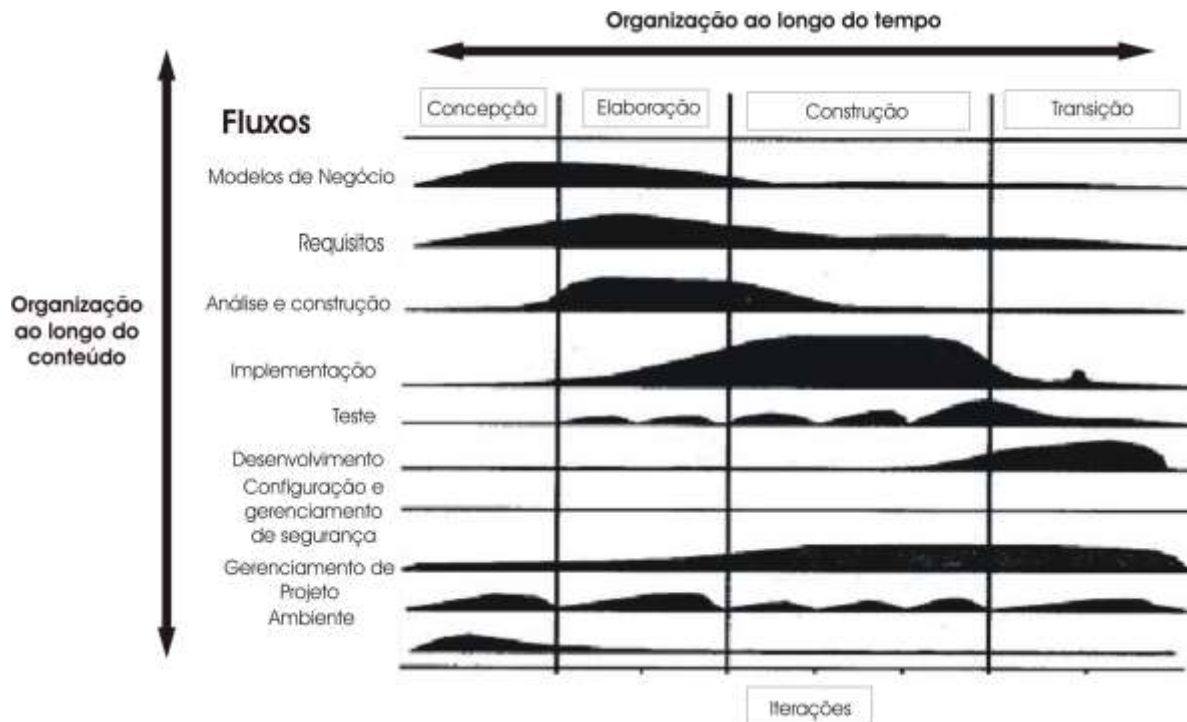


Ilustração 1 – Como o processo é estruturado nos dois eixos. (KRUCHTEN, 2003).

Em primeiro plano temos a representação do aspecto dinâmico do processo: seus ciclos, fases, iterações e marcos. No segundo plano temos o aspecto estático, que abrange sua descrição em relação aos componentes de processo, atividades, fluxos, artefatos e trabalhadores. (PRESMAN, 2006).

O ciclo de vida do software é quebrado em quatro ciclos, cada ciclo gera uma versão do produto. O RUP divide o desenvolvimento nas quatro fases consecutivas abaixo. Quando cada fase é concluída decisões críticas devem ser tomadas e objetivos alcançados (KRUCHTEN, 2003).

- *Concepção (Inception)*
- *Elaboração (Elaboration)*
- *Construção (Construction)*
- *Transição (Transition)*

A concepção (*Inception*) é a primeira fase e esclarece o problema da tecnologia, definindo os casos mais críticos. Definem-se aqui o caso de negócio e escopo do projeto. A Segunda fase é a elaboração (*Elaboration*) que traz uma análise mais detalhada no domínio do problema, desenvolve o projeto, estabelece a arquitetura e também suprime os elementos de alto risco. Na seqüência tem-se a construção (*Construction*), em que ocorrem diversas repetições, sendo que em cada

uma é construído um software de qualidade, o qual passa por uma preparação para chegar aos usuários, produzem-se casos de teste, documentação e código. Finalmente tem-se o quarto e último momento, a transição (*Transition*), na qual os usuários são treinados e o produto passa por uma transição para ser utilizado.

Cada fase do RUP pode, ainda, ser dividida em iterações. Uma iteração é um ciclo completo de desenvolvimento resultando em um produto executável, um subconjunto do produto final ainda em desenvolvimento, que cresce a cada iteração para se tornar a versão final do sistema. O processo iterativo oferece vantagens quando comparado ao processo tradicional de desenvolvimento em cascata com maior reutilização, melhor qualidade geral, mudanças mais facilmente controladas. Até o presente momento a fase do RUP de concepção foi a mais utilizada no trabalho.

O padrão de arquitetura de software aplicado será o MVC, tal padrão é destinado a isolar a interface do usuário da lógica do programa, dividindo o *software* em três partes: o *model* que representa a informação (regra de negócio) da aplicação, o *view* que corresponde a interface do usuário e o *controller* que controla a comunicação entre eles. A utilização deste padrão resulta em um *software* que pode ser modificado mais facilmente, sem que alterações em um parte reflitam em outra.

Neste trabalho, as fases do RUP utilizadas foram: modelagem de negócios, requisitos e análise e projeto. Durante a segunda etapa do trabalho, onde o sistema será desenvolvido, serão também utilizadas as fases implementação e teste.

Posteriormente, nos próximos capítulos, serão abordados respectivamente a modelagem comercial, o fluxo de requisitos, a análise e projeto, as considerações finais, as referências e o apêndice A, neste estarão todos os diagramas desenvolvidos.

2 MODELAGEM COMERCIAL

Citando Kruchten (2003), as metas da modelagem comercial são: definir a estrutura e a dinâmica da organização na qual um sistema será distribuído, entender os problemas atuais na organização alvo e identificar potenciais melhorias, garantir o entendimento comum entre os clientes, usuários finais e desenvolvedores e levantar as exigências de sistema necessárias para o suporte da organização alvo.

Para alcançar tais metas, o fluxo de modelagem comercial descreve como desenvolver uma visão da nova organização alvo, e baseando-se nesta visão, definir os processos, papéis e responsabilidades daquela organização num modelo de negócio (KRUCHTEN, 2003).

1 Introdução

Este documento foi elaborado sobre a norma IEEE std 830 – 1998 e descreve o conteúdo do documento de especificação de exigência de software, apresenta exemplos de possíveis tipos de estruturas. Esta prática pode ser aplicada na assistência de seleção de produtos comerciais.

Este documento é uma recomendação prática da implementação da norma em questão e tem a finalidade de definir as exigências.

1.1 Propósito

O presente documento tem o objetivo de especificar e estabelecer os requisitos para o desenvolvimento de um sistema para demonstração de alguns conhecimentos que foram aprendidos na pós-graduação de desenvolvimento de sistemas com tecnologia Java, visando a obtenção do título deste curso.

1.2 Escopo

Este sistema deve automatizar as funcionalidades do Sismóveis, que deverá disponibilizar ao usuário os itens relacionados a seguir. O sistema não deverá tratar de nenhum aspecto da parte de aluguéis e venda dos imóveis.

- Controle de endereços;
- Controle de imóveis;
- Controle de tipo de imóvel;
- Controle de agendamento de imóvel;
- Controle de Usuários;
- Controle de Usuário Administrador;
- Controle de Usuário Cliente;
- Controle de Usuário Corretor;
- Controle de Status;

1.3 Definições, siglas e abreviações

- **SISIMÓVEIS:** Nome dado ao sistema de agendamento de imóveis.
- **ENDERECOS:** Todos os endereços dos imóveis e dos usuários cadastrados no sistema.
- **IMÓVEIS:** Todos os imóveis cadastrados no sistema.
- **TIPO DE IMÓVEIS:** Todos os tipos de imóveis cadastrados no sistema.
- **AGENDA DE IMÓVEIS:** Representa um imóvel sendo agendado por um usuário cliente, ou usuário corretor.
- **USUÁRIOS:** Todos os usuários cadastrados no sistema.
- **USUÁRIOS CLIENTES:** Todos os usuários clientes cadastrados no sistema.
- **USUÁRIOS CORRETORES:** Todos os usuários corretores cadastrados no sistema.
- **USUÁRIOS ADMINISTRADORES:** Todos os usuários administradores cadastrados no sistema.
- **STATUS:** Todos os status dos clientes cadastrados no sistema.

1.4 Referências

Não aplicável.

1.5 Visão Geral

Não aplicável.

2 Descrição Geral

2.1 Perspectivas do Produto

- O sistema utilizará uma camada de persistência que irá interagir entre o produto e o banco de dados que será o JDBC;
- Requisitos de Software: o sistema será desenvolvido utilizando a linguagem de desenvolvimento JAVA 7 com a IDE(Interface Desktop Enviroment) NetBeans 7.0;
- O sistema será WEB;
- O sistema SISIMÓVEIS é um sistema independente, sem integração com outros sistemas.
- O Sistema utilizará o modelo MVC, que é um padrão de arquitetura de software. Através da utilização deste padrão, serão definidos como os componentes desta aplicação irão interagir ;
- O Sistema utilizará o banco de dados objeto relacional MySQL para persistência das informações;

2.2 Funções do Produto

- Manter dados dos endereços - Incluir / Alterar / Excluir / Consultar
- Manter dados dos imóveis - Incluir / Alterar / Excluir / Consultar
- Manter dados dos tipos de imóveis - Incluir / Alterar / Excluir / Consultar
- Manter dados dos agendamentos dos imóveis - Incluir / Alterar / Excluir / Consultar
- Manter dados dos usuários - Incluir / Alterar / Excluir / Consultar
- Manter dados dos usuários clientes - Incluir / Alterar / Excluir /

Consultar

- Manter dados dos usuários corretores - Incluir / Excluir / Consultar
- Manter dados dos usuários administradores - Incluir / Excluir /

Consultar

- Manter dados dos status dos clientes - Incluir / Alterar / Excluir /

Consultar

2.3 Características do Usuário

O usuário Administrador irá manter os cadastrados operacionais e o cadastro de corretores, já os corretores e clientes vão logar no sistema e farão os agendamentos dos imóveis e deverão ter os seguintes conhecimentos:

- O usuário deve ter treinamento no sistema operacional Windows;
- O usuário deve ter conhecimento na sua área de trabalho (Ramo Imobiliário);
- O usuário deve ter treinamento na operação do SISIMÓVEIS.

2.4 Restrições

Os requisitos mínimos de hardware são: *Intel CORE I3*, HD (*Hard Drive*) de 500 gb, 1 gb de memória RAM (*Random Access Memory*), saída USB (para backup de dados), JVM (*Java Virtual Machine*) instalado, sistema operacional *Microsoft Windows 7*.

2.5 Suposições e Dependências

Não aplicável.

3 Requisitos Específicos

3.1 Requisitos Funcionais

Requisito funcional 1: O sistema deve permitir que os usuários cadastrem os endereços (endereço, cidade, UF e CEP).

Requisito funcional 2: O sistema deve permitir que os usuários

cadastrem os imóveis (Valor do imóvel, quantidade cômodos, tamanho do imóvel, número, complemento).

Requisito funcional 3: O sistema deve permitir que os usuários cadastrem os usuários Administradores(login, senha, nome, número, complemento, email, telefone celular, data do cadastro, renda mensal e ativo).

Requisito funcional 4: O sistema deve permitir que os usuários cadastrem os usuários Clientes(login, senha, nome, número, complemento, email, telefone celular, data do cadastro).

Requisito funcional 5: O sistema deve permitir que os usuários cadastrem os usuários Corretores(login, senha, nome, número, complemento, email, telefone celular, data do cadastro, renda mensal, crd e comissão).

Requisito funcional 6: O sistema deve permitir que os usuário agendem os Imóveis para os clientes e corretores, contendo os dados: Data do agendamento, hora que se inicia o agendamento e a hora que termina o agendamento.

Requisito funcional 7: O sistema deve permitir que os usuários cadastrem os dados do tipo de imóvel, contendo os dados: Descrição do tipo de imóvel.

Requisito funcional 8: O sistema deve permitir que os usuários cadastrem os dados do Status do cliente, contendo os dados: Descrição do status do cliente, e o status do cliente.

3.2 Requisitos Não Funcionais

Requisito não funcional 1: O sistema deve ser totalmente aderente aos principais conceitos de Orientação a Objeto (encapsulamento, herança, polimorfismo, tipificação forte, etc.), visando facilitar a manutenibilidade do mesmo.

Requisito não funcional 2: O sistema deve apresentar interface de fácil utilização e entendimento, permitindo usabilidade.

Requisito não funcional 3: O sistema deve apresentar portabilidade, podendo ser executado em diversos ambientes.

4 FLUXO DE REQUISITOS

O objetivo do fluxo de requisitos é descrever o que o sistema deve fazer e permitir que desenvolvedores e usuários concordem com o que está descrito. Para alcançar tal objetivo é produzida uma declaração delimitada do escopo do problema e do produto, uma lista dos atores identificados no sistema, uma lista de requisitos (organizada por entidades) e um conjunto de cenários de utilização que fornece entendimento do uso do sistema ou do produto. (PRESSMAN, 2006; KRUTCHEN, 2003).

4.1 Diagrama de Casos de Uso

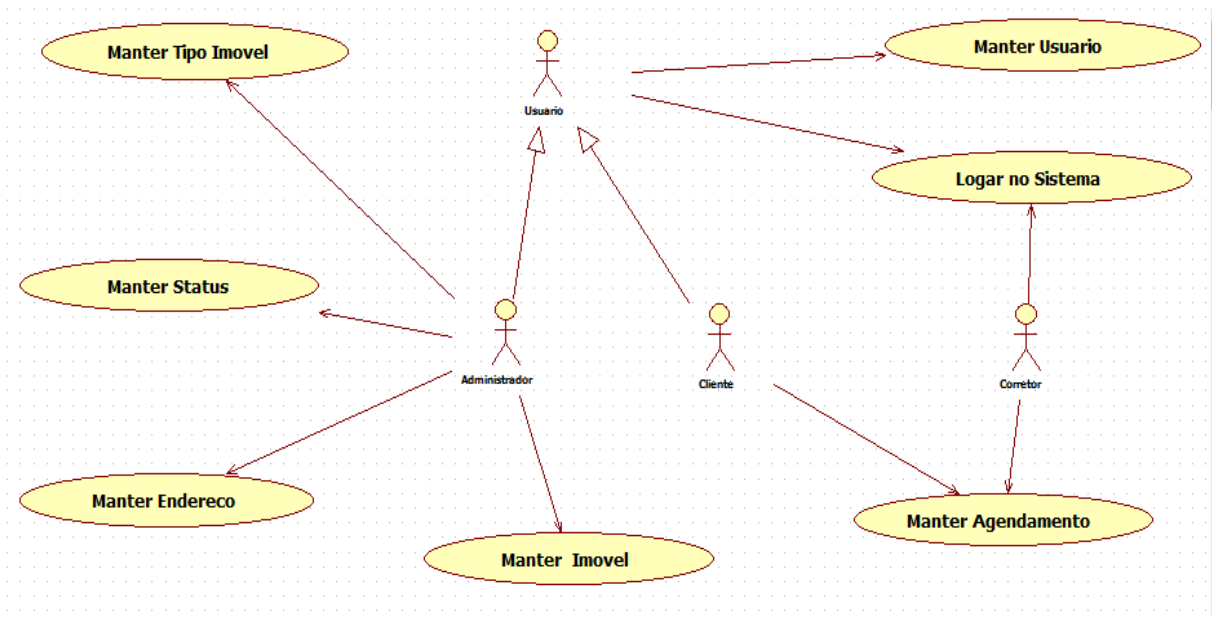


Ilustração 4.1 – Diagrama de Casos de uso do SISIMÓVEIS

O diagrama de caso de uso representa graficamente esta interação, representando os atores como uma figura humana, enquanto os casos de uso são elipses com seus respectivos nomes. A comunicação dos atores com os casos de uso é representada por uma linha unindo os dois elementos.

Na Ilustração 4.1 são apresentados os casos de uso que demonstram as funções que o usuário pode exercer no sistema, seguida pela descrição de um caso de uso. Neste capítulo será descrito o caso de uso Manter Imóveis, contendo toda sua descrição, também serão apresentados o diagrama de caso de uso e o diagrama de seqüência. As outras descrições de casos de uso serão apresentadas no Apêndice A.

4.2 Caso de uso Manter Imóveis

Nome do caso de uso:

Manter Dados Imóveis

Ator principal:

Funcionário

Ator Secundário:

Não aplicável

Descrição:

Este caso de uso permite a manutenção dos dados (inclusão, exclusão, alteração e consulta) dos imóveis.

Pré-Condições

Dados dos endereços e tipos de imóveis mantidos no sistema.

Pós-Condições:

Dados dos imóveis mantidos no sistema.

Fluxo Principal

1. O usuário solicita ao sistema a manutenção de imóveis.
2. OSD apresentar ao usuário a tela de manutenção de imóveis.
3. OSD apresentar ao usuário uma lista de imóveis cadastrados, contendo os dados: Valor, quantidade de cômodos, tamanho, número e complemento. (E1).
4. OSD permitir ao usuário selecionar a opção desejada. (Incluir, Alterar, Excluir e Consultar) (A1)
5. O usuário seleciona a opção desejada.
6. OSD executar o subfluxo correspondente ao tipo de operação recebido (Incluir, Alterar, Excluir, Consultar)

Subfluxos:**Operação Consultar**

7. O usuário seleciona a empresa desejada.
8. O usuário seleciona a opção “Consultar”. (A1)
9. O sistema recupera os dados (Valor, quantidade de cômodos, tamanho, número e complemento.) do imóvel desejado (E1).
10. OSD exibir dados do imóvel selecionado com os campos preenchidos e desabilitados.

Operação Incluir

7. O usuário seleciona a opção “Incluir”.
8. OSD exibir os campos vazios e habilitados.
9. OSD permitir ao usuário informar os dados do imóvel: Valor, quantidade de cômodos, tamanho, número e complemento (V1, V2, V3, A1)
10. O usuário seleciona a opção “Gravar”.
11. OSD gerar o código do imóvel.
12. OSD persistir os dados do imóvel (E2, RN1).
13. OSD apresentar mensagem ao usuário informando o sucesso da realização da operação.
14. OSD recarregar a lista dos imóveis cadastrados.

Operação Alterar

7. O usuário seleciona a empresa desejada.
8. O usuário seleciona a opção “Alterar”.
9. O sistema recupera os dados (Valor, quantidade de cômodos, tamanho, número e complemento) do imóvel desejado (E1).
10. OSD exibir dados do imóvel selecionada com os campos preenchidos e habilitados. (E1)
11. OSD permitir ao usuário alterar os dados do imóvel: Valor, quantidade de cômodos, tamanho, número e complemento. (V1, V2, V3)
12. O usuário seleciona a opção “Gravar”.
13. OSD persistir os dados do imóvel (E2, RN1).
14. OSD apresentar mensagem ao usuário informando o sucesso da realização da operação.
15. OSD recarregar a lista de imóveis cadastrados.

Operação Excluir

7. OSD permitir ao usuário selecionar um imóvel. (A1)
8. O usuário seleciona a opção “Excluir”.
9. OSD confirmar a operação.
10. OSD excluir os dados do imóvel. (E3, RN2)
11. OSD apresentar mensagem ao usuário informando o sucesso da realização da operação.
12. OSD recarregar a lista de imóveis cadastrados.

Requisitos não funcionais:

Não Aplicável

Fluxos de Exceção:

E1. O sistema não consegue recuperar os dados de um imóvel.

1. OSD exibir mensagem de erro.

E2. O sistema não consegue armazenar/atualizar os dados de um imóvel.

1. OSD exibir mensagem de erro.

E3. O sistema não consegue excluir os dados de um imóvel.

1. OSD exibir mensagem de erro.

Fluxos Alternativos:

A1. O usuário cancela a operação.

O sistema deve limpar os campos da interface.

O fluxo retorna ao passo quatro do fluxo principal.

Regras de negócio:

RN1: Os imóveis não podem apresentar o valor zerado.

RN2: Os imóveis devem ter todos os campos preenchidos.

RN3: Um imóvel agendado não pode ser excluído.

Validações

V1: O campo valor não pode ser vazio, e deve ter um valor numérico com casas decimais.

V2: O campo cômodos, metragem e número deve ter um valor numérico .

V3: O campo complemento não pode ser vazio.

5. ANÁLISE E PROJETO

O objetivo da análise e projeto é mostrar o que o sistema realizará na fase de implementação. O comportamento e as funcionalidades do software a ser desenvolvido devem ser definidos e demonstrados, para melhor entendimento do problema. Os diagramas de classe de entidades descrevem as classes, que possuem nome, atributos e operações, tais classes formam a estrutura do sistema e suas relações. (PRESSMAN, 2006).

O principal diagrama dessa fase é o diagrama de classes, que objetiva auxiliar no processo de modelagem e na futura estruturação do código, mas em um nível mais alto. As informações de níveis mais baixos são representadas em outros diagramas, como os diagramas de sequência.

5.1 Diagrama de classes

O diagrama de classes descreve as classes que formam a estrutura do sistema e suas relações, as classes possuem nome, atributos e operações que desempenham no sistema.

Na Ilustração 5.1, a seguir, são ilustradas as classes que compõem o sistema e através dela é visto como estas se relacionam. Existem 9 classes no total, as classes estão descritas com seus respectivos atributos e métodos.

- A classe Imóvel está relacionada com um Endereço e um tipo de imóvel;
- a classe endereço está relacionada com um imóvel e um usuário;
- A classe endereço está relacionada a um ou mais imóveis, e a um ou mais usuários;
- O usuário está relacionado a um administrador, cliente ou a um corretor e a um endereço;
- O usuário administrador possui um relacionamento com o usuário.
- O usuário Cliente possui um relacionamento com o usuário, usuário corretor e a Agenda Imóvel e com Status.
- A classe Corretor possui um relacionamento com o usuário, usuário cliente e com a Agenda Imóvel.
- A classe Status possui um relacionamento com o usuário cliente.

- A classe Tipo de Imóvel possui um relacionamento com o Imóvel.

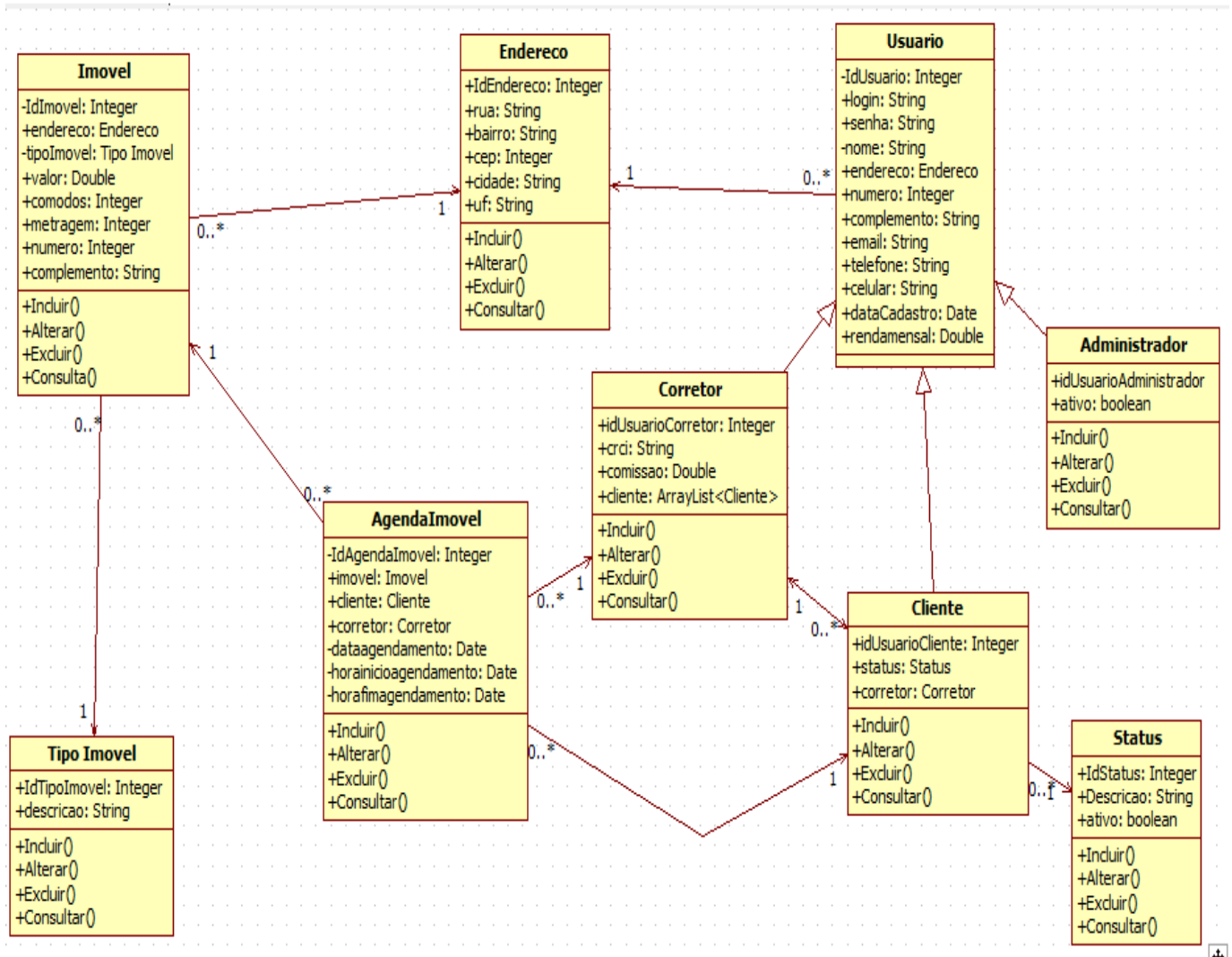


Ilustração 5.1 – Diagrama de Classes de Entidade

5.2 Especificação do Caso de Uso Manter Imóvel

Segundo Fowler (2005), os diagramas de sequência permitem a visualização temporal dos diferentes eventos que ocorrem no sistema. Os objetos estão na parte superior e estes trocam mensagens entre si, as linhas observadas, que quando são cheias indicam uma mensagem de chamada, as tracejadas indicam uma mensagem de retorno. Os diagramas de sequência do caso de uso Manter Empresas serão apresentados agora, os demais virão no apêndice A.

5.3 Diagramas de Sequência

Nesta ordem, serão apresentados os Diagramas de Sequência: Incluir Imóveis, Excluir Imóveis, Alterar Imóveis e Consultar Imóveis.

5.3.1 Diagrama de Sequência Incluir Imóveis

A seguir estão enumerados os eventos ocorridos no diagrama de sequência Incluir Imóveis, Ilustração 5.3 a seguir.

- O usuário solicita a manutenção de Imóveis.
- O sistema deve, então, exibir a tela implementada na classe FrmManter Imovel.
- Esta classe envia a mensagem lista Imóvel () para a classe CtrManter Imovel.
- Esta envia a mensagem lista Imovel () para a classe Endereço.
- Esta envia a mensagem listaEndereco() para a classe Endereço.
- Esta envia a mensagem listaTipolmovel() para a classe Tipo de Imóvel.
- Seguidamente, a classe Imóvel retorna os dados solicitados para a classe CtrlImovel, que por sua vez a encaminha para a classe FrmManterImovel.
- A lista de empresas é exibida ao usuário.
- Este, então, solicita a opção incluir.
- O usuário informa os dados e solicita inclusão dos mesmos, recebendo uma mensagem de confirmação.

- A classe FrmManterImovel valida os campos e envia a mensagem incluir() a classe CtrManterImovel.
- Esta executa o método verificaImovel(Imovel).
- Finalmente, esta classe envia a mensagem incluir() para a classe Imovel, persistindo os dados.
- Esta envia a mensagem listarImovel() para a classe Imovel.

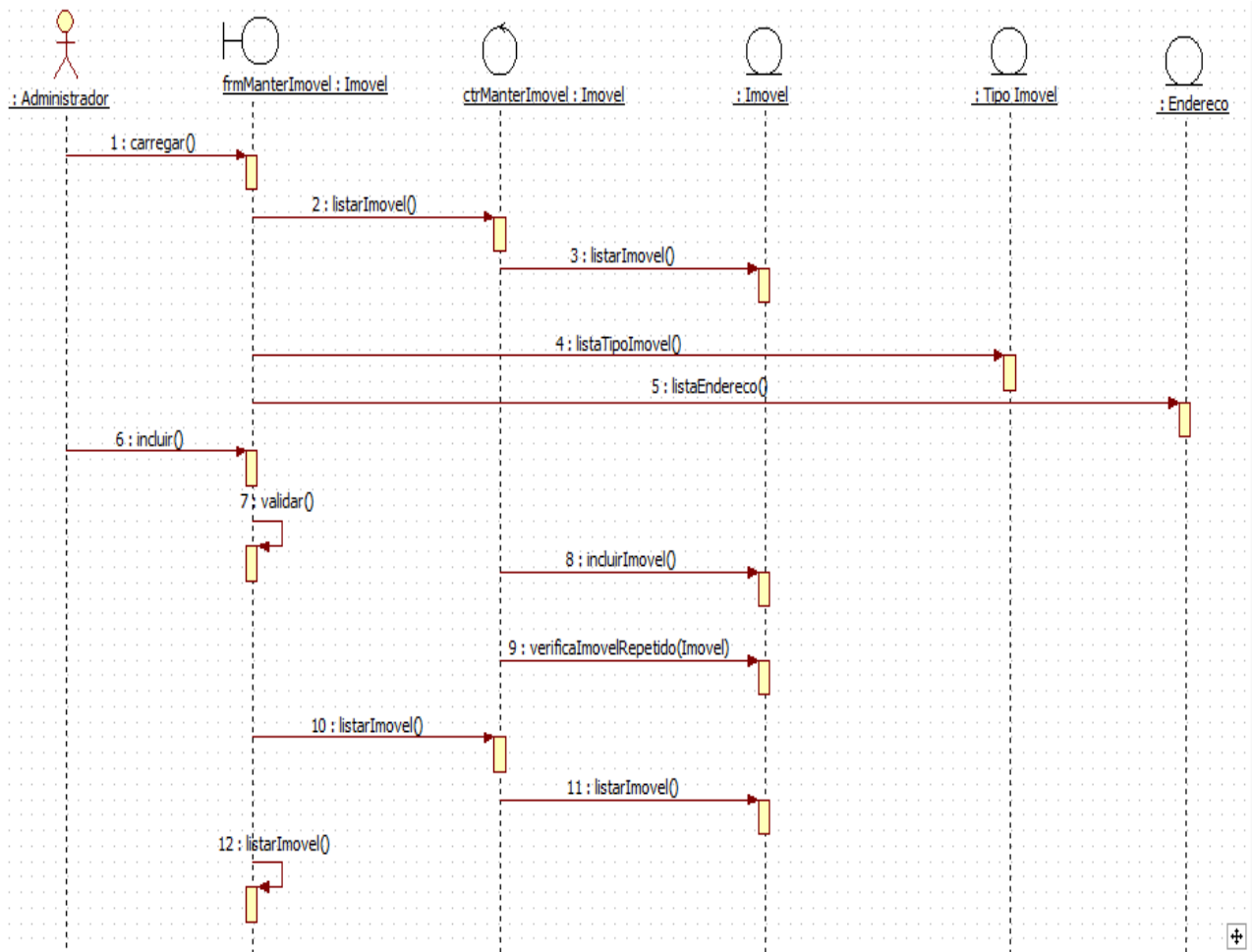


Ilustração 5.3 – Diagrama de Sequência Incluir Imóvel

5.3.2 Diagrama de Sequência Excluir Imóveis

A seguir estão enumerados os eventos ocorridos no diagrama de sequência Excluir Imóveis, Ilustração 5.3.2 a seguir.

- O usuário solicita a manutenção de Imóveis.
- O sistema deve, então, exibir a tela implementada na classe FrmManterImovel.
- Esta classe envia a mensagem listImovel () para a classe CtrManterImovel.
- Esta envia a mensagem listImovel () para a classe Imovel.
- Esta envia a mensagem listaAgendaImovel () para a classe Agenda Imovel.
- Seguidamente, a classe Imovel retorna os dados solicitados para a classe CtrImovel, que por sua vez a encaminha para a classe FrmManterImovel.
- A lista de empresas é exibida ao usuário.
- Este, então, solicita a opção excluir.
- O sistema recupera os dados e solicita exclusão dos mesmos, recebendo uma mensagem de confirmação. A classe FrmManterImovel exibe os dados e envia a mensagem excluirlImovel(Imovel) a classe CtrManterImovel. Esta executa o método excluirlImovel(Imovel). Esta classe envia a mensagem excluirlImovel(Imovel) para a classe Imovel, persistindo os dados. Finalmente, esta classe envia a mensagem excluirlImovelAgendaImovel(Imovel) para a classe Agenda, persistindo os dados. Esta envia a mensagem listImovel() para a classe Imovel.

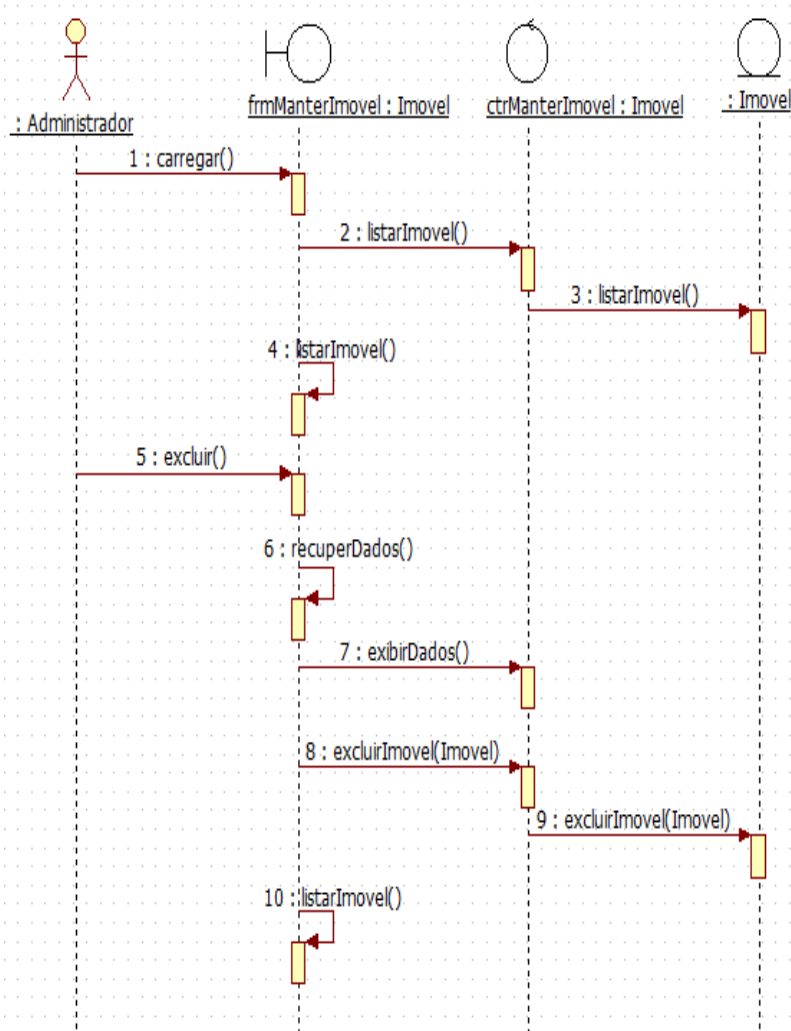


Ilustração 5.3.2 – Diagrama de Sequência Excluir Imóvel

5.3.3 Diagrama de Sequência Alterar Imóveis

A seguir estão enumerados os eventos ocorridos no diagrama de sequência Alterar Imóveis, Ilustração 5.3.3 a seguir.

- O usuário solicita a manutenção de Imóveis.
- O sistema deve, então, exibir a tela implementada na classe FrmManterImovel.
- Esta classe envia a mensagem listImovel() para a classe CtrManter Imovel.
- Esta envia a mensagem lista Imovel para a classe Empresa.
- Esta envia a mensagem listaEndereco() para a classe Endereco.
- Esta envia a mensagem listaTipoImovel() para a classe Tipo de Imóvel.
- Seguidamente, a classe Imovel retorna os dados solicitados para a classe CtrlImovel, que por sua vez a encaminha para a classe FrmManter Imovel.
- A lista de Imoveis é exibida ao usuário.
- Este, então, solicita a opção alterar.
- O usuário informa os dados e solicita alteração dos mesmos, recebendo uma mensagem de confirmação.
- A classe FrmManterImovel valida os campos e envia a mensagem alterar() a classe CtrManterImovel.
- Esta executa o método verificImovel (Imovel).
- Finalmente, esta classe envia a mensagem alterar() para a classe Imovel, persistindo os dados.
- Esta envia a mensagem listImovel () para a classe Imovel.

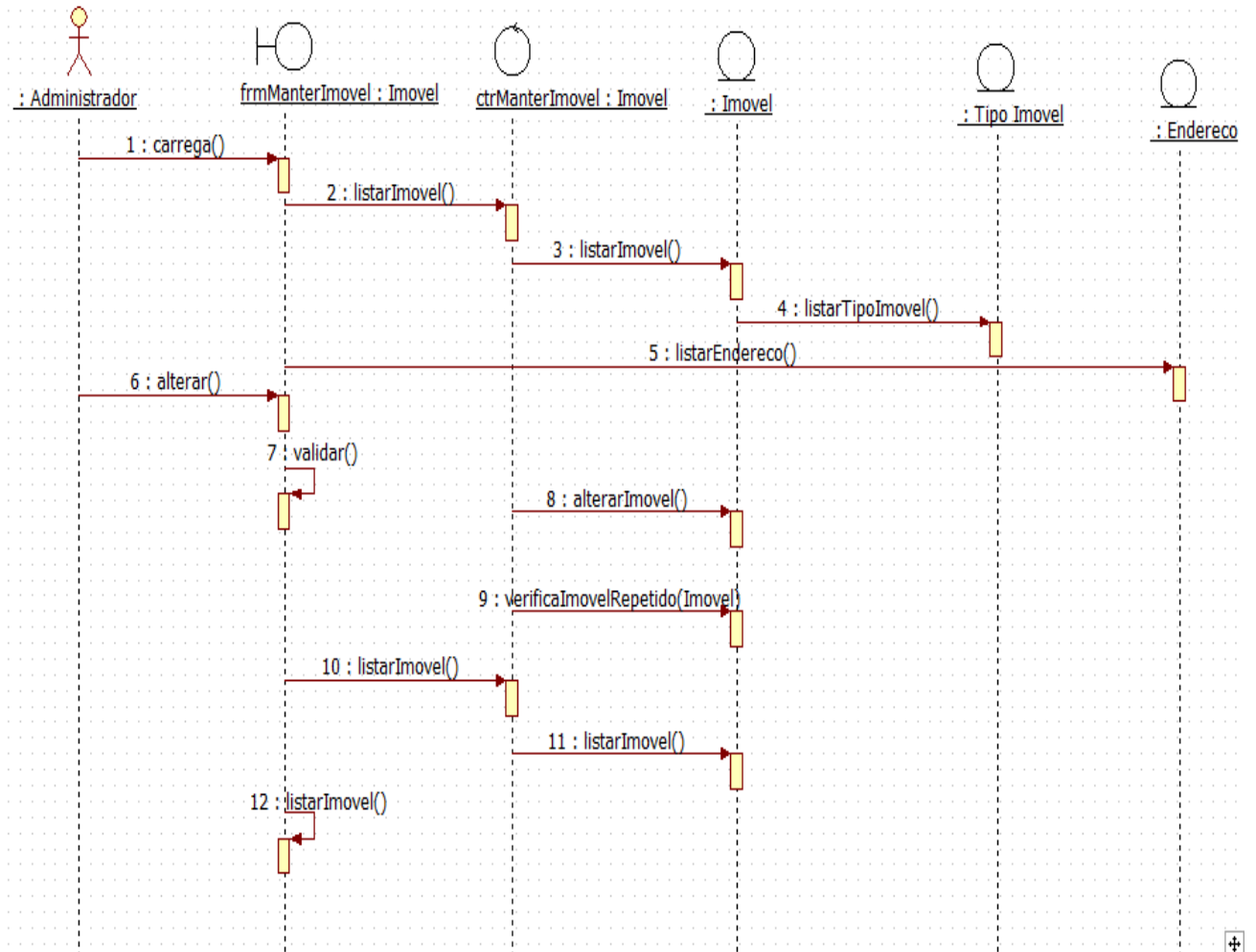


Ilustração 5.3.3 – Diagrama de Sequência Alterar Imóvel

5.3.4 Diagrama de Sequência Consultar Imóvel

A seguir estão enumerados os eventos ocorridos no diagrama de sequência Consultar Imóvel, Ilustração 5.3.4 a seguir.

- O usuário solicita a manutenção de Imóveis.
- O sistema deve, então, exibir a tela implementada na classe FrmManterImovel.
- Esta classe envia a mensagem listImovel() para a classe CtrlManterImovel.
- Esta envia a mensagem listImovel () para a classe Empresa.
- Esta envia a mensagem listaEndereco() para a classe Endereço.
- Este envia a mensagem listaTipoImovel() para a classe Tipo de imovel
- Seguidamente, a classe Imóvel retorna os dados solicitados para a classe CtrlImovel, que por sua vez a encaminha para a classe FrmManterImovel.
- A lista de imóveis é exibida ao usuário.
- Este, então, solicita a opção consultar.
- A classe FrmManterImovel recupera os dados.
- A classe FrmManterImovel exibe os dados.

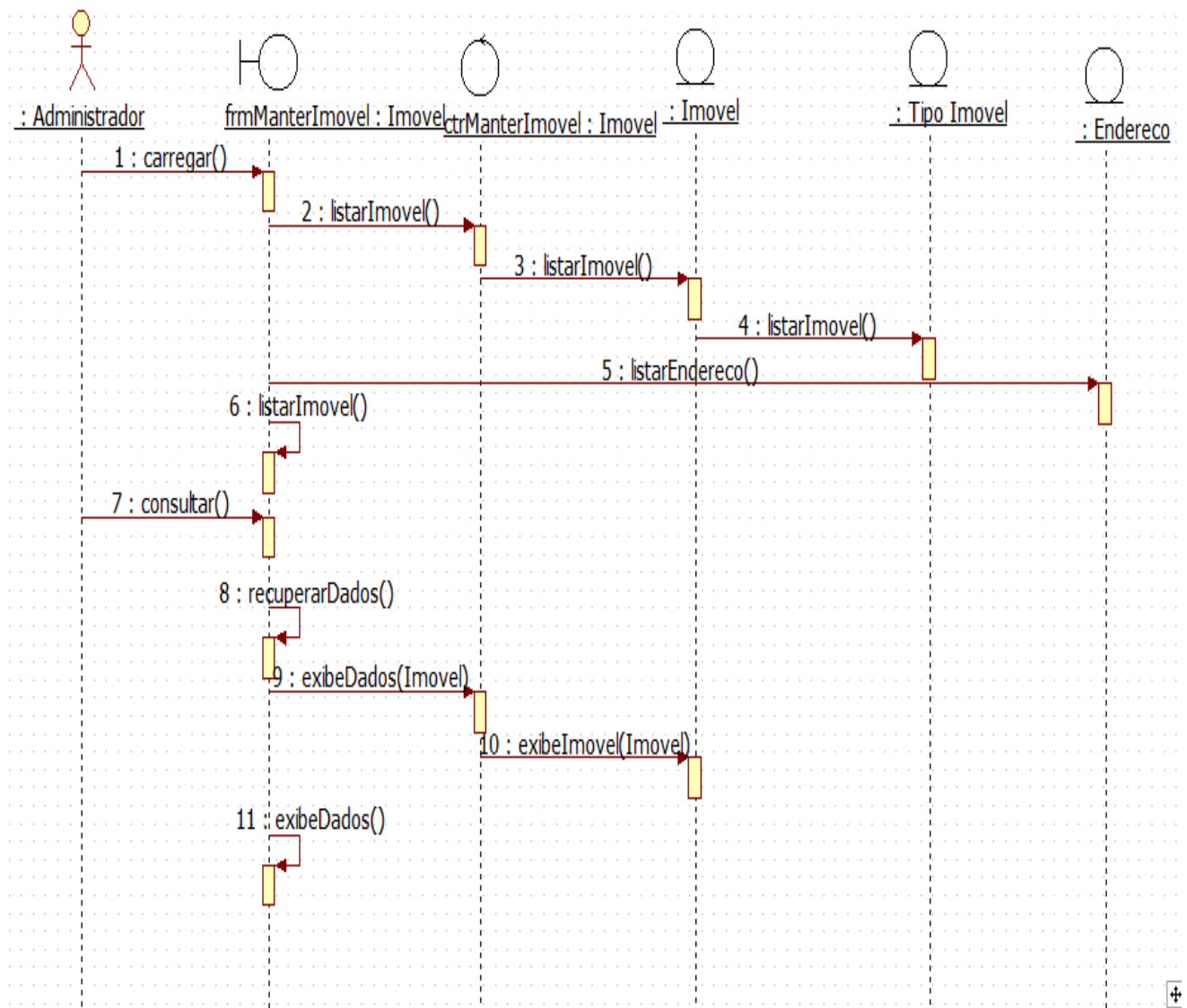


Ilustração 5.3.4 – Diagrama de Sequência Consultar Imóvel

5.4.1 Padrões de projetos.

Os padrões de projeto (Design Patterns) oferecem um meio para nomear e descrever soluções abstratas para problemas recorrentes.

Em outras palavras, padrões de projeto são soluções que projetistas experientes já utilizaram anteriormente, empregadas com sucesso para a solução de problemas específicos de projetos, tornando os projetos orientados a objetos mais flexíveis e, em última instância, reutilizáveis (GAMMA et al., 2000).

As duas palavras que compõem este método de desenvolvimento de aplicação padrão e projeto (design) merecem atenção. Define-se padrão, como uma forma ou modelo proposto para imitação e um sistema coerente compreensível baseado nas inter-relações pretendidas entre as partes componentes.

Além disso, define-se design como: Planejamento com propósitos determinados e um projeto mental ou esquema no qual os meios para alcançar um objetivo são estabelecidos;

Uma das chaves do desenvolvimento orientado ao objeto, de fato qualquer desenvolvimento, é sua reutilização. A maioria do trabalho da reutilização está concentrado na reutilização do código. Os padrões de projeto permitem que se reutilizem os designs. Desta forma, pode-se criar uma interface padrão dos componentes, reutilizando-os quando necessário sempre que uma interface semelhante estiver disponível.

Neste trabalho é utilizado o padrão de projeto Singleton. Este padrão de projeto tem a função de garantir que uma classe tenha somente uma instância e fornecer um ponto global de acesso para a mesma (GAMMA et al., 2000). Um dos módulos que utiliza este padrão é o modulo de conexão com o banco de dados, para que se mantivesse a conexão até o fim da utilização do sistema, evitando uma nova conexão a cada solicitação ao servidor. Deste modo, obtém-se uma diminuição do tráfego de informações junto ao servidor, criando uma única conexão (caso esta não esteja criada) ou retornando a conexão ativa.

5.4.2 O padrão model-view-controller (MVC)

Segundo PRESSMAN (2006), o objetivo do padrão de arquitetura MVC é separar a lógica de (Modelo) da interface do usuário (Visão) e do fluxo de aplicação (Controlador). A idéia é permitir que uma mesma lógica de negócios possa ser acessada através de várias interfaces utilizando-se de alguns padrões de projeto para definir como os componentes da aplicação irão interagir entre si.

A idéia básica da arquitetura MVC é minimizar a ligação entre objetos em um sistema alinhando-os com um conjunto específico de responsabilidades na área dos dados permanentes e regras associadas (Model), apresentação (View) e a lógica da aplicação (Controller). Na arquitetura MVC, a lógica de negócios não sabe de quantas nem quais interfaces com o usuário estão exibindo seu estado, conforme é demonstrado na Ilustração 5.4.2

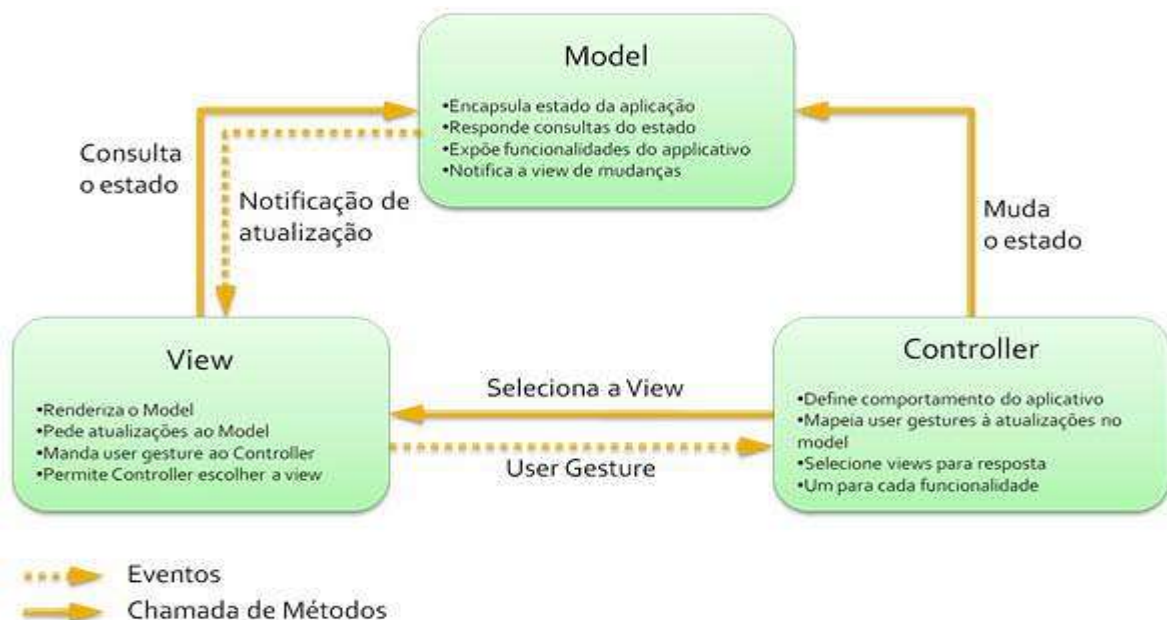


Ilustração 5.4.2 – MVC Model 2 proposto pela (SUN, 2004)

Neste padrão, a aplicação é dividida em três partes:

- modelo (MODEL): lógica de negócio ou aplicação;
- visão (VIEW): camada de interface com o usuário. Nesta camada o usuário vê o estado do modelo e pode manipular a interface, para ativar a lógica do negócio;
- controlador (CONTROLLER): transforma eventos gerados pela

interface em ações de negócio, alterando o modelo.

O Model é responsável por manter o estado e os dados da aplicação. Pode receber e responder as consultas a partir de View e pode fornecer notificações para View quando algum dado é modificado ou um evento é disparado.

O Controller atualiza o Model com base na execução da lógica da aplicação em resposta aos gestos do usuário (por exemplo, botões da caixa de diálogo, solicitações de envio do formulário, etc.). Também é responsável por informar a View o que exibir em resposta aos gestos do usuário.

O View é responsável pela apresentação real dos dados fornecidos por Controller, bem como a manipulação e criação de todos os objetos que são exibidos para o usuário.

5.4.3 Arquitetura em camadas.

Dando continuidade aos conceitos do padrão MVC, observou-se que nos sistemas onde não há uma divisão entre a interface do sistema com o usuário, a forma de como os dados são validados, persistência e recuperação dos dados, pode existir um alto grau de acoplamento entre os objetos e classes, contribuindo para que estes sistemas possuam baixa manutenibilidade e dificultando a sua reutilização em outros projetos (FRAGMENTAL, 2008).

Para contornar este problema, devem-se agrupar as responsabilidades do sistema de acordo com um critério, diminuindo o acoplamento entre os componentes e fazendo com que as mudanças em um grupo não impactem muito em outro. Para realizar esta separação, a Arquitetura de Camadas é muito utilizada (FRAGMENTAL, 2008).

Neste trabalho, os módulos foram divididos em utilizando camadas, que são (Ilustração 5.4.3).

A camada de apresentação é responsável pelos objetos que o usuário pode visualizar na tela na tela e pela sua interação com o sistema e a camada de negócio,

ela contém as regras de negócio da aplicação.

A camada de persistência contém as classes que possibilitarão uma interação entre o sistema e o banco de dados.

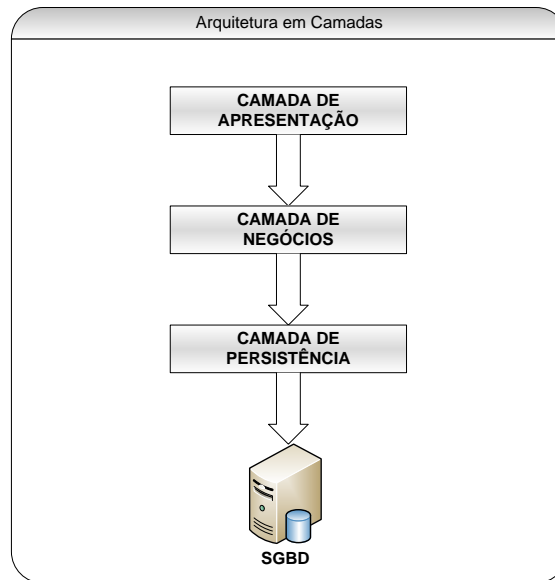


Ilustração 5.4.3 – Arquitetura em camadas

Segundo a Sun (2008), a separação da camada de negócios e a apresentação trazem uma série de benefícios, dentre eles: minimização do impacto das mudanças: a camada de negócios pode ser alterada sem que sejam necessárias muitas modificações na camada de apresentação.

Em Alguns casos, não é necessário alterar a camada de apresentação. A facilidade de manutenção: como as camadas são divididas em componentes de negócios distintos, as regras podem ser acessadas a partir de vários objetos distintos na aplicação, e seu código-fonte concentrado em um só local. como resultado, tem-se um código-fonte mais enxuto e o mesmo comportamento em todo lugar em que estes forem utilizados. caso haja a necessidade de uma alteração nas regras de negócio, basta alterar o arquivo que contém estas regras. benefícios similares também são alcançados na camada de apresentação, como a reutilização de folhas de estilo, padrões de formatação, etc;

Reutilização de código: como cada módulo do sistema está adequadamente dividido de acordo com um critério e de forma independente, respeitando os conceitos de orientação a objetos e encapsulamento, estes módulos são facilmente integrados

em outros projetos;

Especialização: como o sistema está dividido em camadas com critérios distintos, a equipe de desenvolvimento pode ser dividida de acordo com estes interesses. Um exemplo deste benefício são os responsáveis pelo design e interface do sistema, que terão acesso somente ao código necessário para desenvolver o design e a interface do sistema.

Como formas de armazenamento de informações no sistema são utilizadas bases de dados. A maioria das instituições que utilizam um sistema de banco de dados são completamente dependentes desta tecnologia. A tecnologia de banco de dados relacional é muito conhecida e, além disso, é flexível e possui grande técnica de gerenciamento de dados (BAUER, 2005).

É possível desenvolver aplicativos utilizando o paradigma da orientação a objeto utilizando a linguagem SQL para acessar e persistir os dados, embora esta forma de desenvolvimento requeira muito tempo e esforço. Segundo Bauer, até 30% do código escrito do aplicativo é desenvolvido para o acesso aos dados via SQL. Ainda assim, o resultado não é muito confiável.

Para realizar a integração entre os dados do sistema e os dados do banco de dados, serão empregados objetos de acesso, conhecidos como DAO (Data Access Object), que permitem a divisão das regras de negócio e das regras para acesso aos dados. Para isso, criam-se classes exclusivas para o acesso ao SGBD, evitando o acoplamento de códigos e comandos na linguagem SQL com os componentes de negócio (XAVIER, 2007).

6 IMPLEMENTAÇÃO

6.1 Visão geral

A implementação deste trabalho foi realizada na IDE(Integrated Development Environment) NetBeans 7.1, com a utilização do *framework* JDBC(Java Data Base Connection) para a persistência no banco de dados relacional MySQL, sendo este responsável pelo gerenciamento da base de dados, de forma a gerenciar o acesso, manipulação e organização dos dados. Foi utilizada a abordagem MVC, sendo possível realizar a divisão em pacotes, separando todas as classes de negócio, visão e controle, conforme ilustra a figura a seguir:

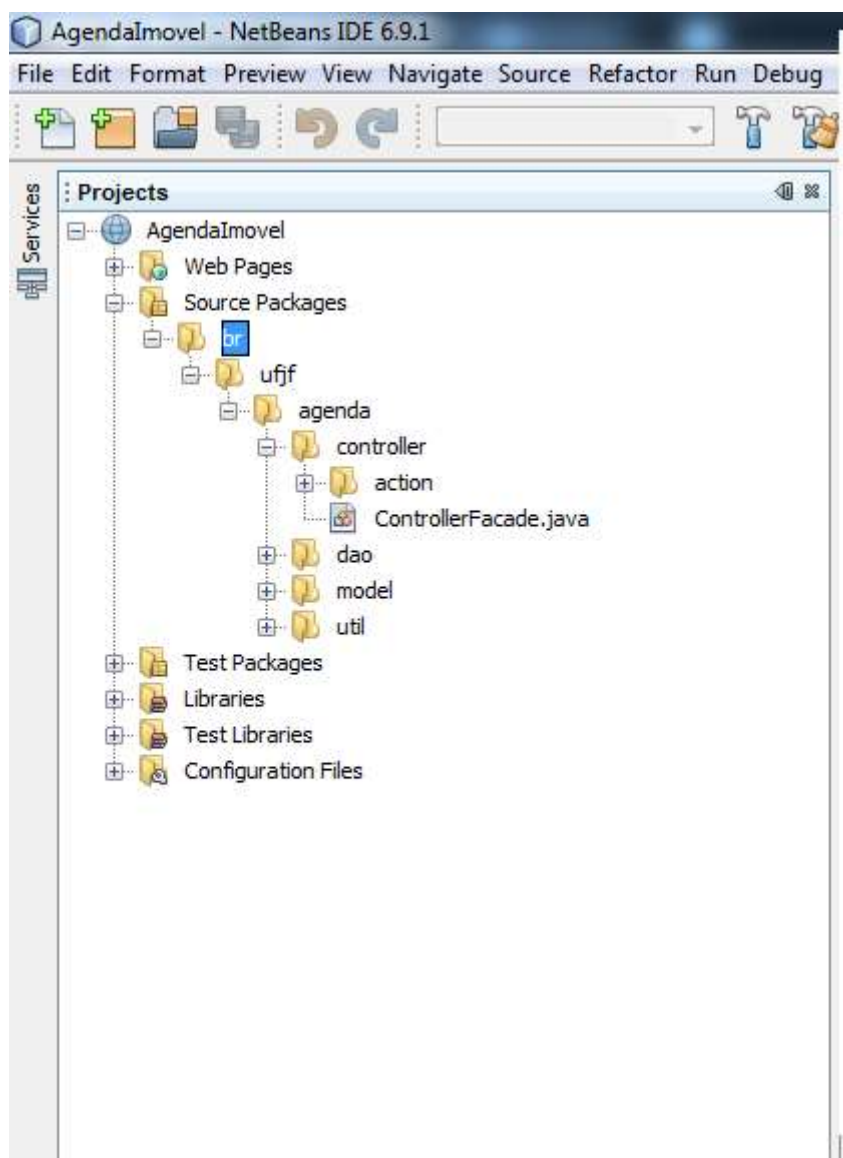


Ilustração 6.1 – Divisão do SISIMOVEIS

Vale ressaltar, que além da separação da visão, controle e DAO foram realizados pacotes específicos para validação, utilidades e imagem, facilitando a construção e futura manutenção do software.

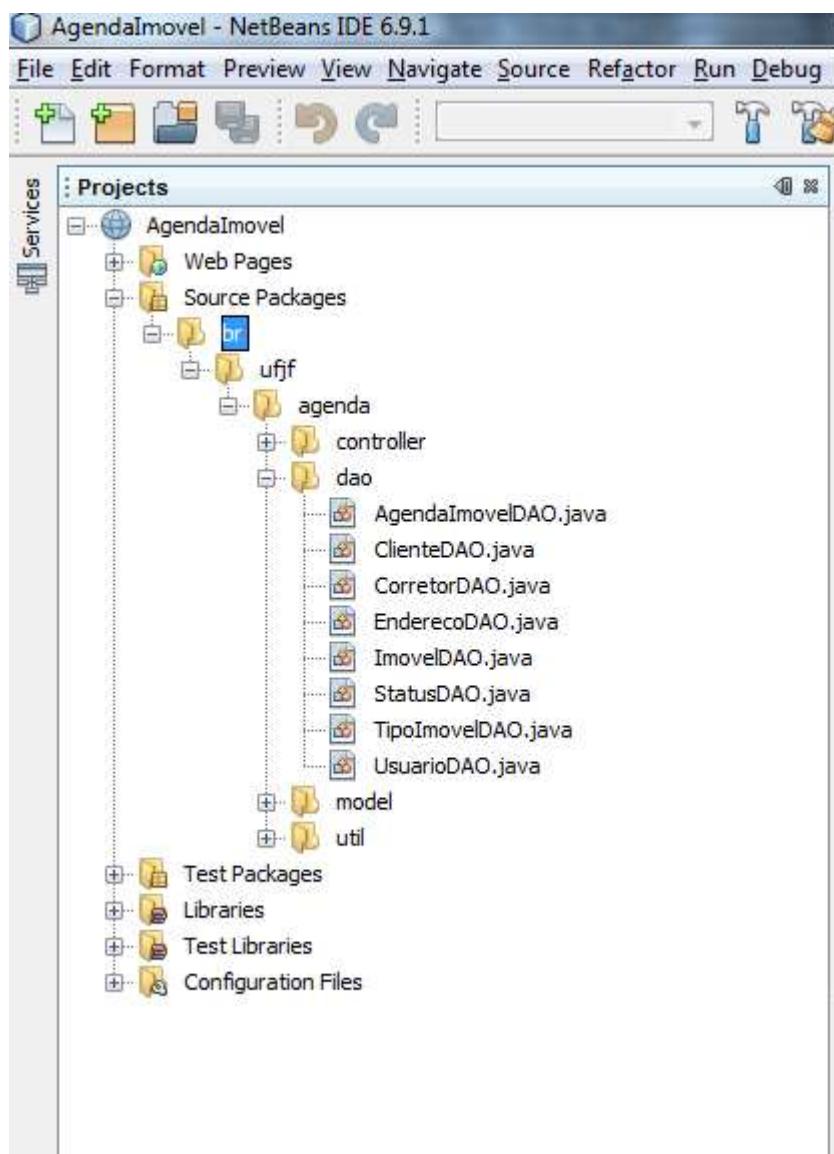


Ilustração 6.1.1 – Divisão das classes DAO.

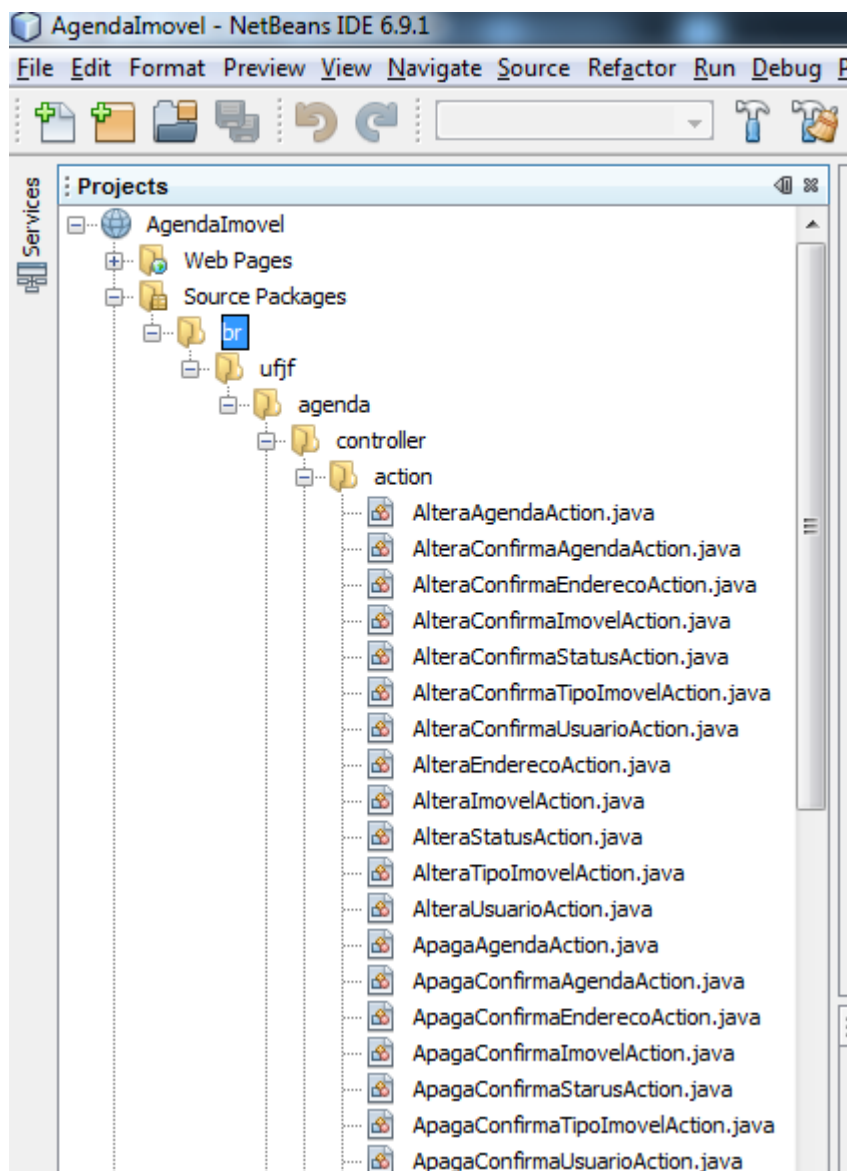


Ilustração 6.1.2 – Divisão das classes Controller.

A Ilustração 6.1.1 e 6.1.2, reflete que para cada controlador presente no pacote Controller é criado uma classe DAO, correspondente ao controlador tratado. Esta por sua vez implementa métodos comuns a estes DAO, como gravar, alterar, excluir, carregar um objeto e recuperar uma lista de objetos pelo atributo estabelecido.

6.2 Aplicação das tecnologias

Neste tópico serão apresentados todas as tecnologias aplicadas na construção do SISIMÓVEIS, desta forma os objetivos propostos por este trabalho foram concluídos.

A seguir, serão descritas as tecnologias JDBC permitindo assim uma melhor compreensão sobre o uso destas tecnologias abordadas no desenvolvimento do SISIMÓVEIS.

6.2.1 JDBC.

Java Data Base Connectivity ou JDBC, é um conjunto de classes ou interfaces (API) escritas em Java que fazem o envio de instruções SQL para qualquer banco de dados relacional. Api de baixo nível e base para api's de alto nível. Amplia o que você pode fazer com Java Possibilita o uso de banco de dados já instalados. Para Cada banco de dados há um driver JDBC que pode cair em quatro categorias.

- Ponte JDBC-ODBC;
- Driver API-Nativo;
- Driver de protocolo de Rede;
- Driver Nativo;

Vale ressaltar que com o uso do JDBC foi possível automatizar muitas tarefas repetitivas de códigos SQL, presentes no desenvolvimento do sistema. O JDBC o manipula de forma consolidada, realizando apenas o esforço de construção e de manutenção dos mapeamentos.

```

5 package br.ufjf.agenda.dao;
6
7 import br.ufjf.agenda.model.Imovel;
8 import br.ufjf.agenda.model.TipoImovel;
9 import br.ufjf.agenda.util.GenericDAO;
10 import java.sql.ResultSet;
11 import java.sql.SQLException;
12 import java.util.ArrayList;
13
14 /**
15  *
16  * @author ice
17  */
18 public class ImovelDAO extends GenericDAO<Imovel, Integer> {
19
20     public void inserir(Imovel obj) throws ClassNotFoundException, SQLException {
21         String sql = "INSERT INTO imovel(idendereco, idtipoimovel, valor, comodos, metragem, numero, complemento) "
22             + "VALUES (?, ?, ?, ?, ?, ?, ?) ";
23
24         ArrayList<String> lista = new ArrayList<String>();
25         lista.add(String.valueOf(obj.getEndereco().getIdendereco()));
26         lista.add(String.valueOf(obj.getTipoImovel().getIdtipoimovel()));
27         lista.add(String.valueOf(obj.getValor()));
28         lista.add(String.valueOf(obj.getComodos()));
29         lista.add(String.valueOf(obj.getMetragem()));
30         lista.add(String.valueOf(obj.getNumero()));
31         lista.add(obj.getComplemento());
32
33         executaDML(sql, lista);
34

```

Ilustração 6.2.1 – Mapeamento da classe Imóvel

A Ilustração 6.2.1, demonstra o mapeamento da classe Imóvel. O relacionamento representa o relacionamento da classe imóvel e a classe tipo de imóvel, onde um tipo de imóvel está relacionado a um ou mais Imóvel, e um Imóvel está relacionado a um Tipo de Imóvel.

6.2.3 Java Data Base Connection.

Com o intuito de aumentar ainda mais a portabilidade do SISIMOVEIS foi usado neste trabalho a Java Data Base Connection, ou simplesmente JDBC, ele é um padrão de programação Java que permite aos desenvolvedores o mapeamento objetos relacional em aplicações que usem Java.

Neste trabalho o JDBC é usado “acima” do controlador, o que facilita uma possível troca, se necessário for.

6.3 Apresentação da aplicação.

Visando facilitar a apresentação do sistema, este item demonstra os passos que este irá realizar na tela manutenção empresa, de forma a explicar as ações que o sistema realiza entre a visão, controlador, modelo e DAO.

Vale ressaltar que as classes utilizadas da visão são FrmSisImoveis e FrmManterImovel, os controladores usado e um deles é ApagarImovelAction, a classe modelo usada será Imovel e o DAO usado será ImovelDao.

O usuário executa a aplicação e está iniciada com a tela principal do sistema, mostrada na Ilustração 6.3.1.

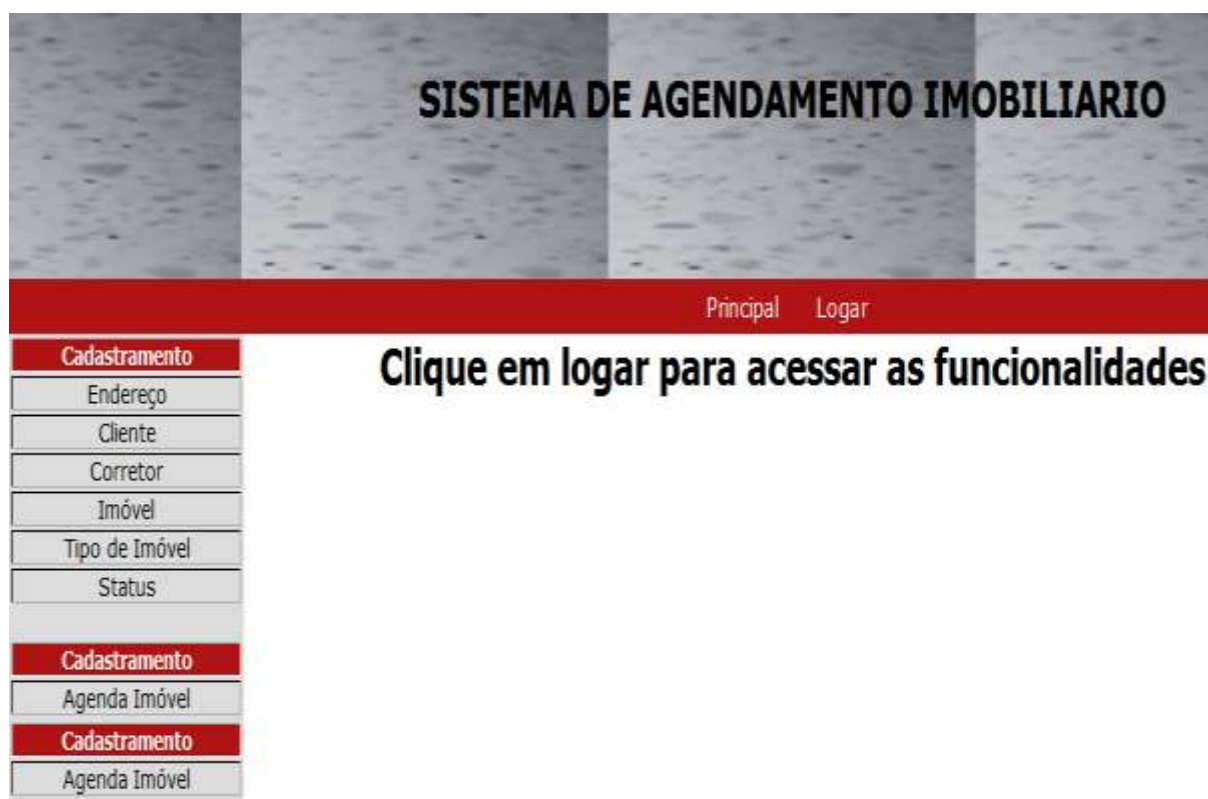


Ilustração 6.3.1 – Tela principal do SisAcoes(FormSisImoveis)

Conforme mostrado na Ilustração 6.3.1 a tela principal do sistema exibe o menu de navegação, onde para efetuar um cadastro de Imóvel o usuário deve clicar em visualizar e depois em Imóvel.

O FormSisImoveis então desencadeia uma ação, instancia o NovoImovel, centraliza-o e o mostra-o na tela, conforme a Ilustração 6.3.2.

SISTEMA DE AGENDAMENTO IMOBILIARIO

Principal Logar

Cadastros

Endereço	Tipo de Imóvel	Valor	Cômodos	Metragem	Número	Complemento	Apagar	Alterar
Batista de Oliveira 45	Casa	200.0	4	3	155	apt 402	X	✎
Sampaio 436	Sítio	500.0	8	120	200	bloco1	X	✎

Nova Agenda

Ilustração 6.3.2 – Tela manter Imovel

O form da Ilustração 6.3.2 quando inicializado, através do método `BuscarTodos()` envia uma mensagem para o controlador do imóvel, requisitando a lista de imóveis cadastrados, este então repassa a requisição para o DAO do imóvel, que retorna uma lista, que por fim é exibida no grid.

Para prosseguir o com a operação desejada o usuário deve clicar então no botão incluir, este então executa o método `NovoImovel()` que instancia o `NovoImovel.JSP`, centraliza-o e mostra-o na tela, conforme a Ilustração 6.3.3

The screenshot displays a web application interface for a real estate scheduling system. The browser window shows the URL `localhost:8080/AgendaImovel/controller?action=NovoImovel`. The page features a red navigation bar with 'Principal' and 'Logar' links. A sidebar on the left contains a menu of buttons: 'Cadastramento' (highlighted), 'Endereço', 'Cliente', 'Corretor', 'Imóvel', 'Tipo de Imóvel', and 'Status'. The main content area contains a form for adding a new property, with fields for 'Endereço:', 'Tipo de Imóvel:', 'Valor:', 'Cômodo:', 'Metragem:', 'Número:', and 'Complemento:'. Below the form are 'Gravar' and 'Voltar' buttons. At the bottom of the sidebar, there are additional buttons for 'Cadastramento' and 'Agenda Imóvel'.

Ilustração 6.3.3 – Tela de inclusão do Imóvel

O usuário deve então preencher os dados e clicar no botão Gravar, que instancia um novo Imóvel e executa o método `NovoImovel()`, que posteriormente valida os campos obrigatórios e manda o objeto Imóvel para o controlador de Imóvel, que posteriormente repassa o objeto para o DAO imóvel que por fim persiste no banco.

Para maiores detalhes sobre o código fonte da aplicação este está disponível no CD que segue em anexo junto ao trabalho, bem como as informações apresentadas no DTR e no Apêndice B, que contém as principais telas do SISIMOVEIS.

7 CONSIDERAÇÕES FINAIS

O presente trabalho tem como principal objetivo o desenvolvimento de um *software* para automatizar uma empresa que agende imóveis para clientes e corretores, que é um sistema de controle de agendamento de imóveis, voltado para *Web*, utilizando a linguagem de programação *Java*.

Ficou claro que, com o desenvolvimento de sistemas utilizando-se de camadas, é possível sua posterior reutilização. Também é notável a maior facilidade em se realizar uma manutenção no sistema, pois o desenvolvedor mantém o foco apenas nos pontos onde realmente ocorrem os problemas, evitando também que o mesmo modifique outras partes do sistema enquanto tenta corrigi-los.

Utilizou-se a ferramenta netbeans 7.1 para o desenvolvimento do sistema e das implementações dos diagramas, possibilitando a utilização por parte do funcionário, utilizando JDBC junto com SQL que permitem acessar informações sem usando SQL e possibilitam facilmente a troca por outro banco de dados sem alterações significativas no código da aplicação.

Finalmente, através deste trabalho foi possível conhecer as características dos frameworks utilizados, de forma a facilitar no desenvolvimento. Estudos futuros serão indispensáveis na aplicação deste trabalho, tais como a geração de gráficos, geração de relatórios, consulta que possam retornar informações gerenciais e a transposição deste aplicativo para um sistema Mobile.

REFERÊNCIAS

KRUCHTEN, P., **Introdução ao RUP Rational Unified Process**, 2. ed., Rio de Janeiro: Ciência Moderna, 2003.

PENDER, T., **UML A Bíblia**, 1. ed., Rio de Janeiro: Campus, 2004.

PRESMAN, R. S., **Engenharia de Software**, 6. ed., São Paulo: McGraw-Hill, 2006.

FOWLER, MARTIN, **UML essencial: um breve guia para a linguagem-padrão de modelagem de objetos**, 3. ed. – Porto Alegre: Bookman, 2005.

BOOCH, GRADY; RUMBAUGH, JAMES; JACOBSON, IVAR. **UML, Guia do Usuário**. 1. ed., Rio de Janeiro: Campus, 2000.

GAMMA, ERICH; HELM, RICHARD; JOHNSON, RALPH; VLISSIDES, JOHN. **Padrões de projeto: soluções reutilizáveis de software orientado a objetos**. Porto Alegre: Bookman, 2000.

MICROSYSTEMS, SUN. **Java EE Application Design With MVC**. Disponível em <<http://java.sun.com/developer/technicalArticles/javase/mvc/>>. Acesso em 26 de maio de 2009.

XAVIER, K.; S., Y.. Abstract factory aplicado. **Java Magazine, Rio de Janeiro, edição 50, p. 40-45, 2007.**

CALÇADO, P., **MVC e camadas**. Disponível em: <http://fragmental.com.br/wiki/index.php?title=MVC_e_Camadas>. Acesso em: 17 out. 2008.

APÊNDICES

Apêndice A – Demais Casos de uso do SISIMOVEIS

O apêndice A contém os demais Casos de Uso com suas respectivas descrições e Diagramas de Sequência.

Caso de uso Manter Endereço.

Nome do caso de uso:

Manter Dados Endereços

Ator principal:

Usuário Administrador

Ator Secundário:

Não aplicável

Descrição:

Este caso de uso permite a manutenção dos dados (inclusão, exclusão, alteração e consulta) dos Endereços.

Pré-Condições:

Não aplicável.

Pós-Condições:

Dados dos Endereços mantidos no sistema.

Fluxo Principal

1. O usuário solicita ao sistema a manutenção de endereços.
2. OSD apresentar ao usuário a tela de manutenção de endereços.
3. OSD apresentar ao usuário uma lista de ações cadastradas, contendo os dados: Endereço, bairro, cidade, UF e CEP. (E1).
4. OSD permitir ao usuário selecionar a opção desejada. (Incluir, Alterar, Excluir e Consultar) (A1)
5. O usuário seleciona a opção desejada.

6. OSD executar o subfluxo correspondente ao tipo de operação recebida (Incluir, Alterar, Excluir, Consultar)

Subfluxos:

Operação Consultar

7. O usuário seleciona o endereço desejado.
8. O usuário seleciona a opção “Consultar”. (A1)
9. O sistema recupera os dados (Endereço, bairro, cidade, UF e CEP) do Endereço desejado (E1).
10. OSD exibir dados do endereço selecionado com os campos preenchidos e desabilitados.

Operação Incluir

7. O usuário seleciona a opção “Incluir”.
8. OSD exibir os campos vazios e habilitados.
9. OSD permitir ao usuário informar os dados do endereço: Endereço, bairro, cidade, UF e CEP, selecionando o endereço. (V1, V2, V3, A1)
10. O usuário seleciona a opção “Gravar”.
11. OSD gerar o código do endereço.
12. OSD persistir os dados do endereço (E2, RN1).
13. OSD apresentar mensagem ao usuário informando o sucesso da realização da operação.
14. OSD recarregar a lista de endereços cadastrados.

Operação Alterar

7. O usuário seleciona o endereço desejado.
8. O usuário seleciona a opção “Alterar”.
9. O sistema recupera os dados (Endereço, bairro, cidade, UF e CEP) do endereço desejado (E1).
10. OSD exibir dados do endereço selecionado com os campos preenchidos e habilitados. (E1)
11. OSD permitir ao usuário alterar os dados do endereço: Endereço, bairro, cidade, UF e CEP, selecionando o endereço. (V1, V2, V3)
12. O usuário seleciona a opção “Gravar”.
13. OSD persistir os dados do endereço (E2, RN1).
14. OSD apresentar mensagem ao usuário informando o sucesso da realização da operação.
15. OSD recarregar a lista de endereços cadastrados.

Operação Excluir

7. OSD permitir ao usuário selecionar um endereço. (A1)
8. O usuário seleciona a opção “Excluir”.
9. OSD confirmar a operação.
10. OSD excluir os dados do endereço. (E3)
11. OSD apresentar mensagem ao usuário informando o sucesso da

realização da operação.

12. OSD recarregar a lista de endereços cadastrados.

Requisitos não funcionais:

Não Aplicável

Fluxos de Exceção:

E1. O sistema não consegue recuperar os dados de um endereço.

1. OSD exibir mensagem de erro.

E2. O sistema não consegue armazenar/atualizar os dados de um endereço.

1. OSD exibir mensagem de erro.

E3. O sistema não consegue excluir os dados de um endereço.

1. OSD exibir mensagem de erro.

Fluxos Alternativos:

A1. O usuário cancela a operação.

O sistema deve limpar os campos da interface.

O fluxo retorna ao passo quatro do fluxo principal.

Regras de negócio:

RN1: As ações têm obrigatoriamente uma empresa.

Validações

V1: O campo endereço não pode ser vazio.

V2: O campo bairro não pode ser vazio.

V3: Os campos cidade, uf e cep não podem ser inválidos.

Diagrama de Sequência Incluir Endereço

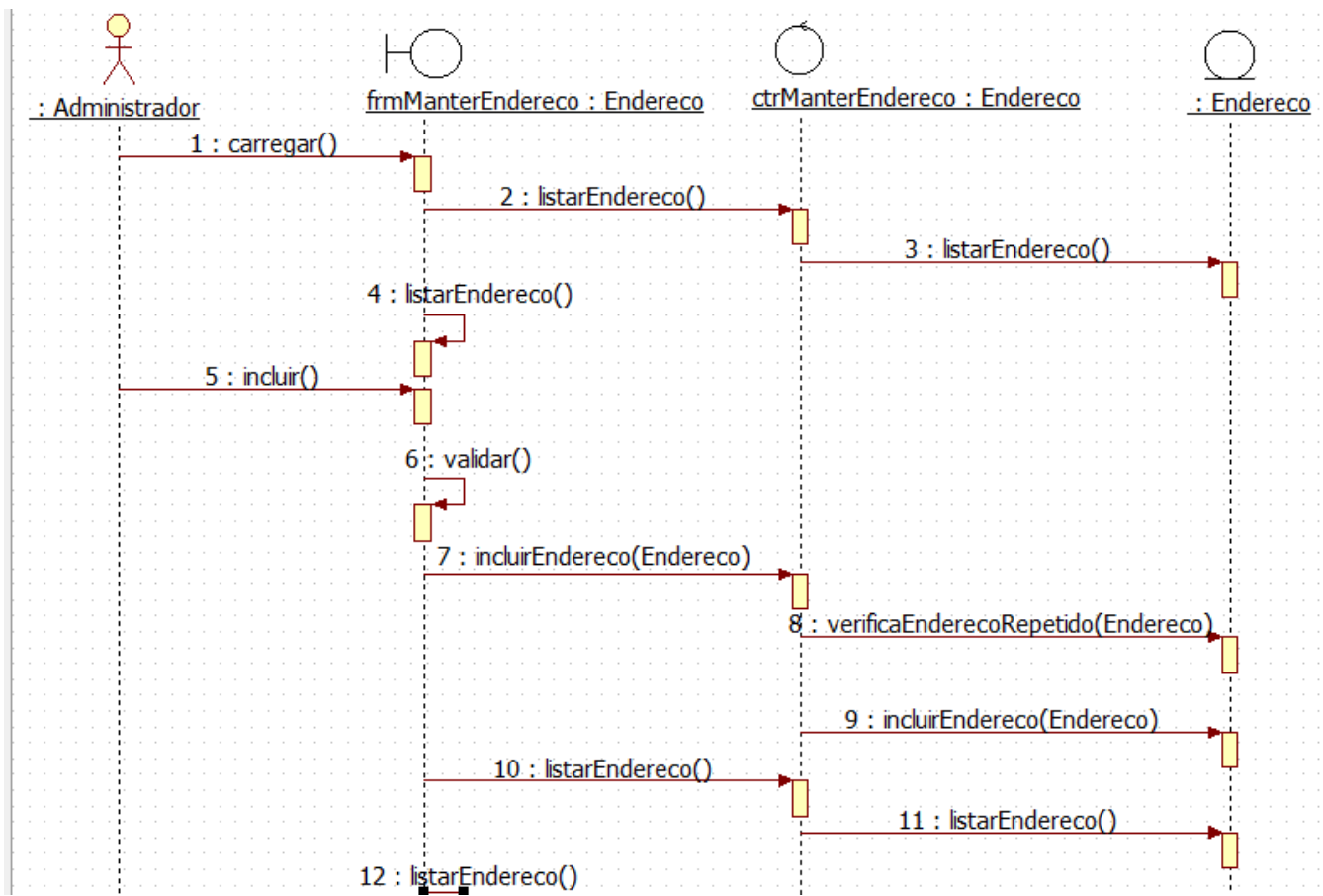


Ilustração A. 1 – Diagrama de sequência incluir endereços

Diagrama de Sequência Excluir Endereço

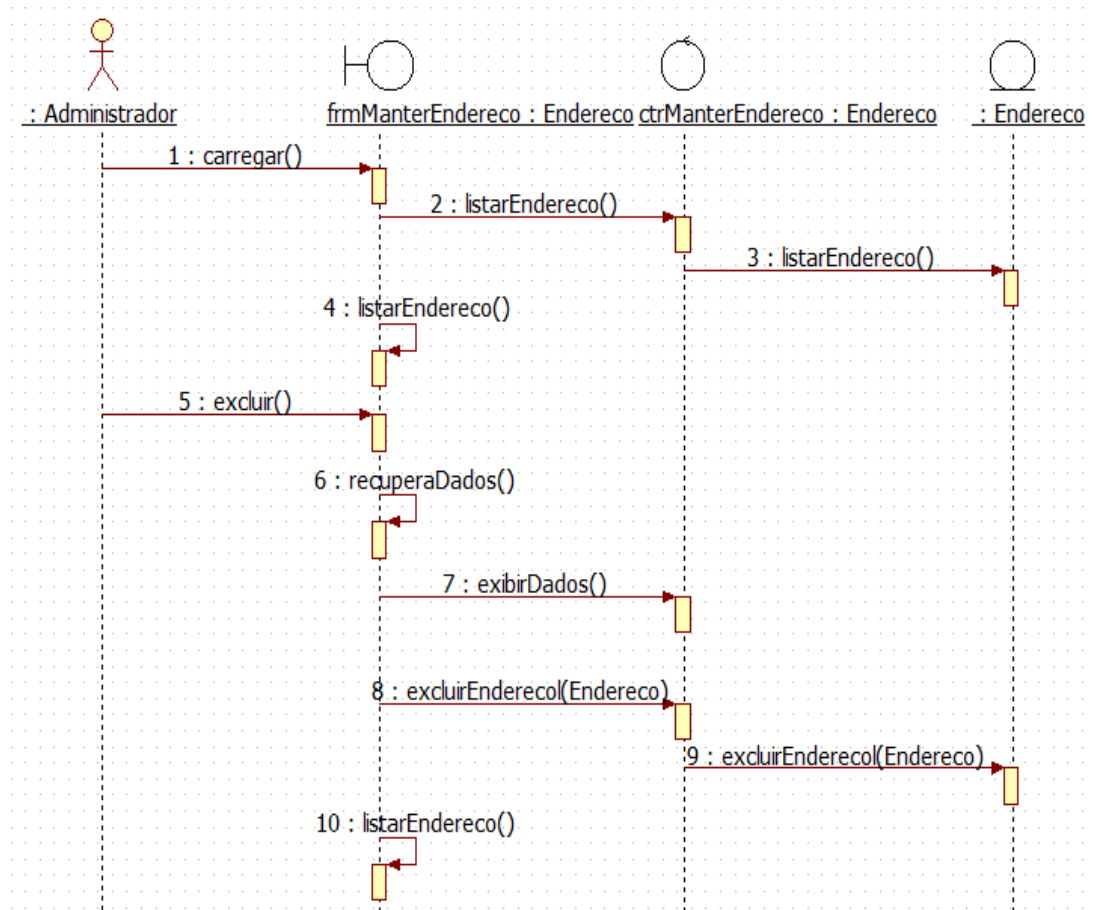


Ilustração A. 2 – Diagrama de sequência excluir endereços

Diagrama de Sequência Alterar Endereço

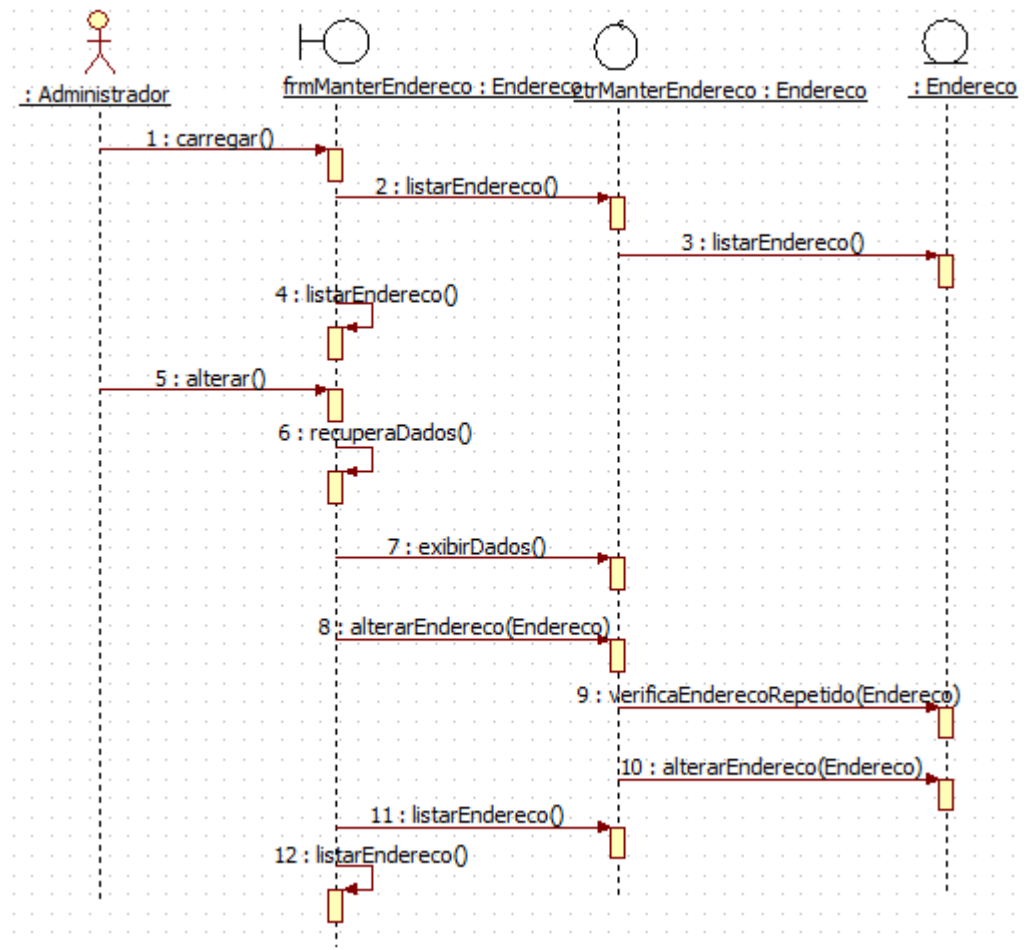


Ilustração A. 3 – Diagrama de sequência alterar endereços

Diagrama de Sequência Consultar Endereço

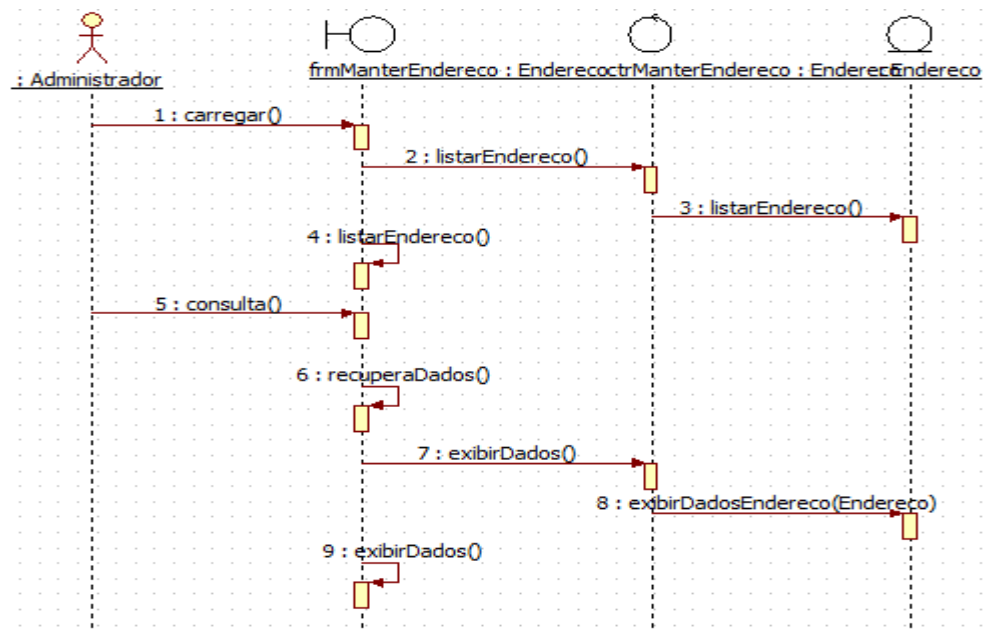


Ilustração A. 4 – Diagrama de sequência consultar endereços

Caso de uso Manter Usuários.

Nome do caso de uso:

Manter Dados Usuários

Ator principal:

Usuário Administrador

Ator Secundário:

Não aplicável

Descrição:

Este caso de uso permite a manutenção dos dados (inclusão, exclusão, alteração e consulta) dos usuários.

Pré-Condições:

Dados do endereço mantidos no sistema.

Pós-Condições:

Dados dos usuários mantidos no sistema.

Fluxo Principal

- 1.O usuário administrador solicita ao sistema a manutenção de usuários.
- 2.OSD apresentar ao usuário administrador a tela de manutenção de usuários.
- 3.OSD apresentar ao usuário administrador uma lista de usuários cadastrados, contendo os dados: login, senha, nome, numero, complemento, email, telefone, celular, data do cadastro e renda mensal. (E1).
- 4.OSD permitir ao usuário administrador selecionar a opção desejada. (Incluir, Alterar, Excluir e Consultar) (A1)
- 5.O usuário administrador seleciona a opção desejada.
- 6.OSD executar o subfluxo correspondente ao tipo de operação recebida (Incluir, Alterar, Excluir, Consultar)

Subfluxos:

Operação Consultar

7. O usuário administrador seleciona o usuário desejado.
8. O usuário administrador seleciona a opção "Consultar". (A1)

9. O sistema recupera os dados (nome, numero, complemento, email, telefone, celular, data do cadastro e renda mensal) do usuário desejado (E1).
10. OSD exibir dados do usuário selecionado com os campos preenchidos e desabilitados.

Operação Incluir

7. O usuário administrador seleciona a opção “Incluir”.
8. OSD exibir os campos vazios e habilitados.
9. OSD permitir ao usuário administrador informar os dados do usuário: login, senha, nome, numero, complemento, email, telefone, celular, data do cadastro e renda mensal. (V1, V2, V3, A1)
10. O usuário administrador seleciona a opção “Gravar”.
11. OSD gerar o código do usuário.
12. OSD persistir os dados do usuário (E2, RN1).
13. OSD apresentar mensagem ao usuário administrador informando o sucesso da realização da operação.
14. OSD recarregar a lista de usuários cadastrados.

Operação Alterar

7. O usuário administrador seleciona o usuário desejado.
8. O usuário administrador seleciona a opção “Alterar”.
9. O sistema recupera os dados (nome, numero, complemento, email, telefone, celular, data do cadastro e renda mensal) do usuário desejado (E1).
10. OSD exibir dados do usuário selecionado com os campos preenchidos e habilitados. (E1)
11. OSD permitir ao usuário alterar os dados do usuário: senha, nome, numero, complemento, email, telefone, celular, data do cadastro e renda mensal. (V1, V2, V3)
12. O usuário administrador seleciona a opção “Gravar”.
13. OSD persistir os dados do usuário (E2, RN1).
14. OSD apresentar mensagem ao usuário administrador informando o sucesso da realização da operação.
15. OSD recarregar a lista de usuários cadastrados.

Operação Excluir

7. OSD permitir ao usuário administrador selecionar um usuário. (A1)
8. O usuário seleciona a opção “Excluir”.
9. OSD confirmar a operação.
10. OSD excluir os dados do usuário. (E3)
11. OSD apresentar mensagem ao usuário administrador informando o sucesso da realização da operação.
12. OSD recarregar a lista de usuários cadastrados.

Requisitos não funcionais:

Não Aplicável

Fluxos de Exceção:

E1. O sistema não consegue recuperar os dados de um usuário.

1. OSD exibir mensagem de erro.

E2. O sistema não consegue armazenar/atualizar os dados de um usuário.

1. OSD exibir mensagem de erro.

E3. O sistema não consegue excluir os dados de um usuário.

1. OSD exibir mensagem de erro.

Fluxos Alternativos:

A1. O usuário cancela a operação.

O sistema deve limpar os campos da interface.

O fluxo retorna ao passo quatro do fluxo principal.

Regras de negócio:

RN1: Os usuários têm obrigatoriamente um endereço.

Validações

V1: Os campos login, senha e nome não podem ser vazios.

V2: Os campos número, complemento e email não podem ser vazios.

V3: Os campos telefone, celular, data de cadastro e renda mensal não podem ser vazios.

Diagrama de Sequência Incluir Usuários.

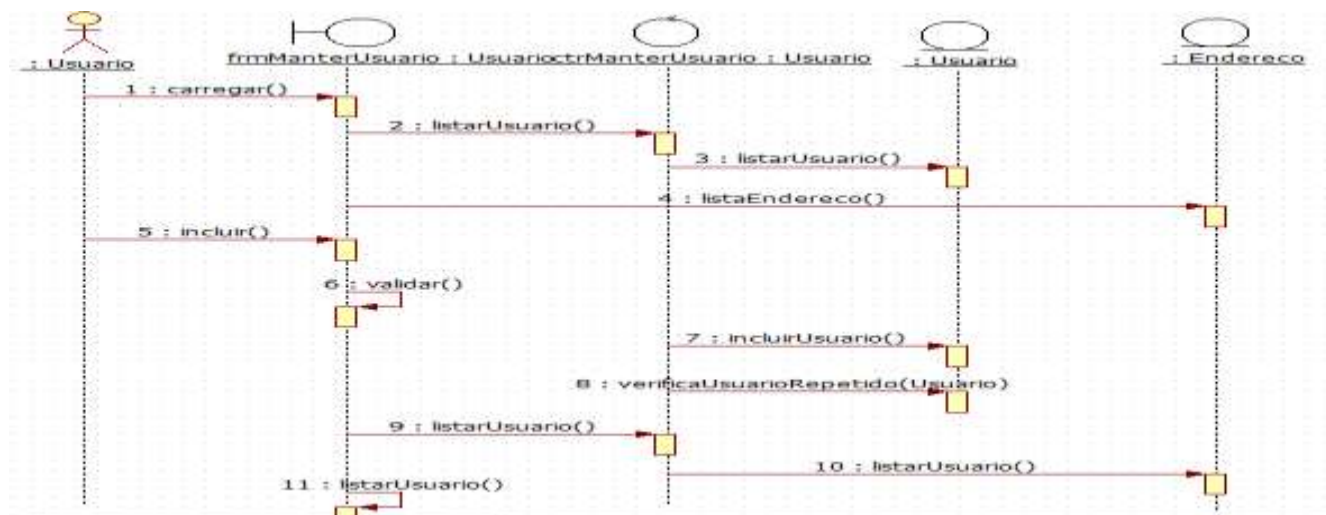


Ilustração A. 5 – Diagrama de sequência incluir usuários.

Diagrama de Sequência Excluir Usuários

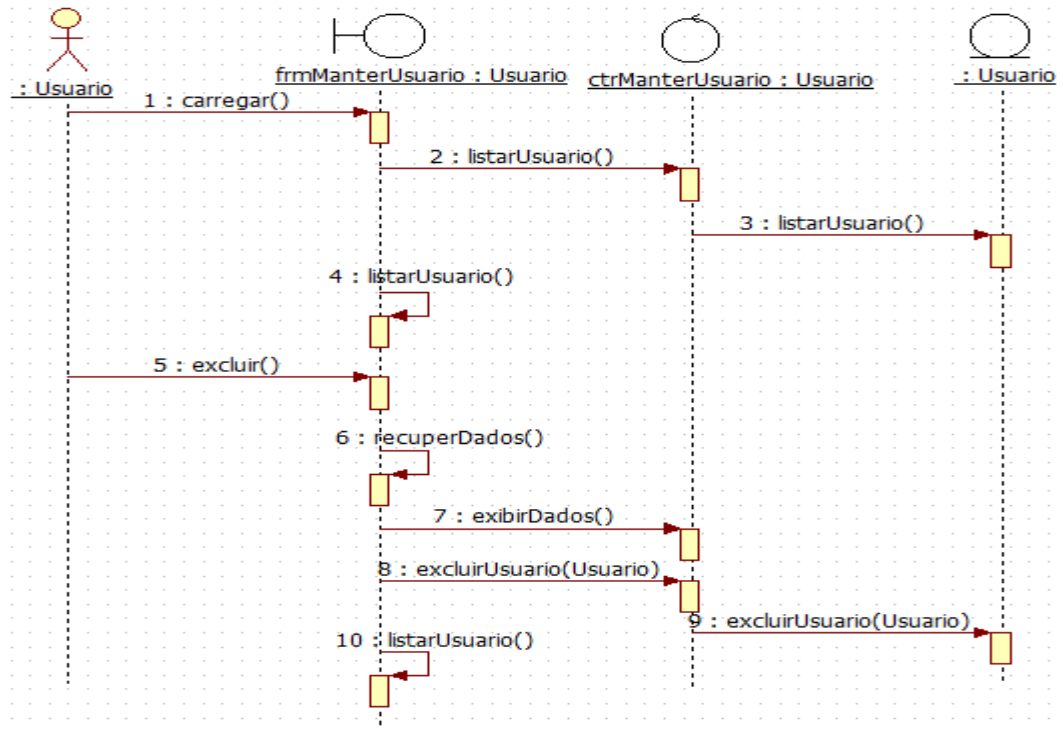


Ilustração A. 6 – Diagrama de sequência excluir usuários

Diagrama de Sequência Alterar Usuários.

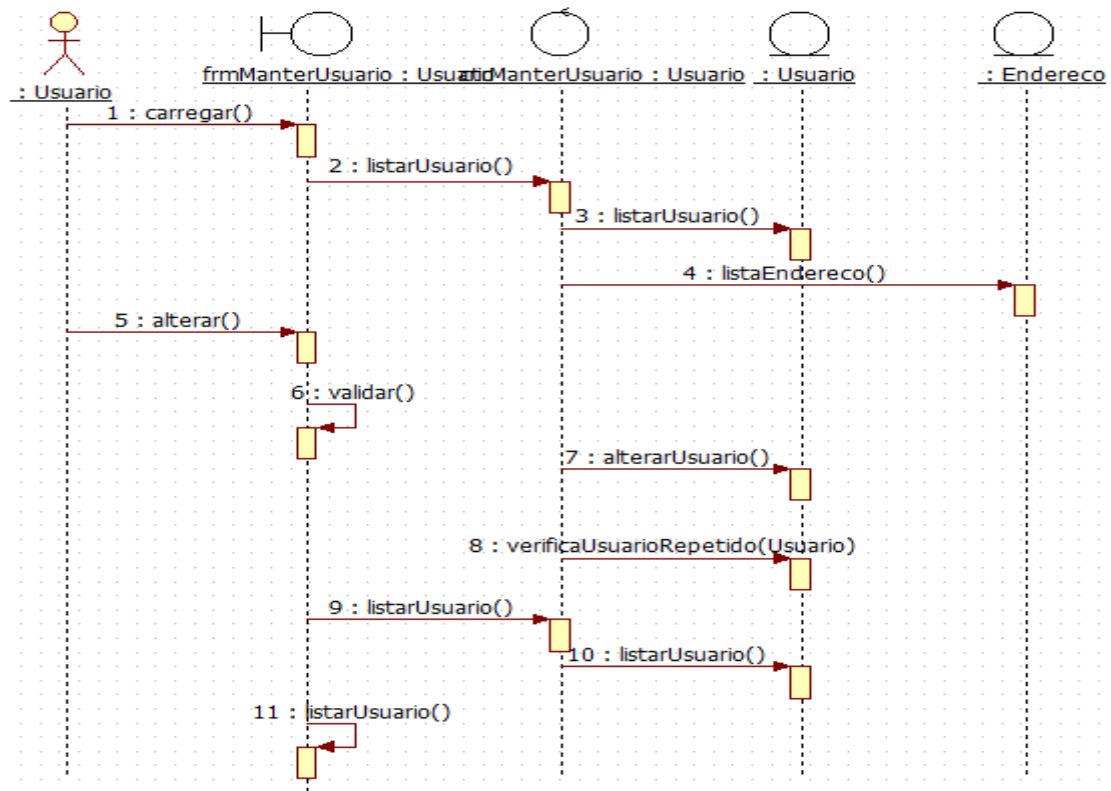


Ilustração A. 7 – Diagrama de sequência alterar usuários

Diagrama de Sequência Consultar Usuários

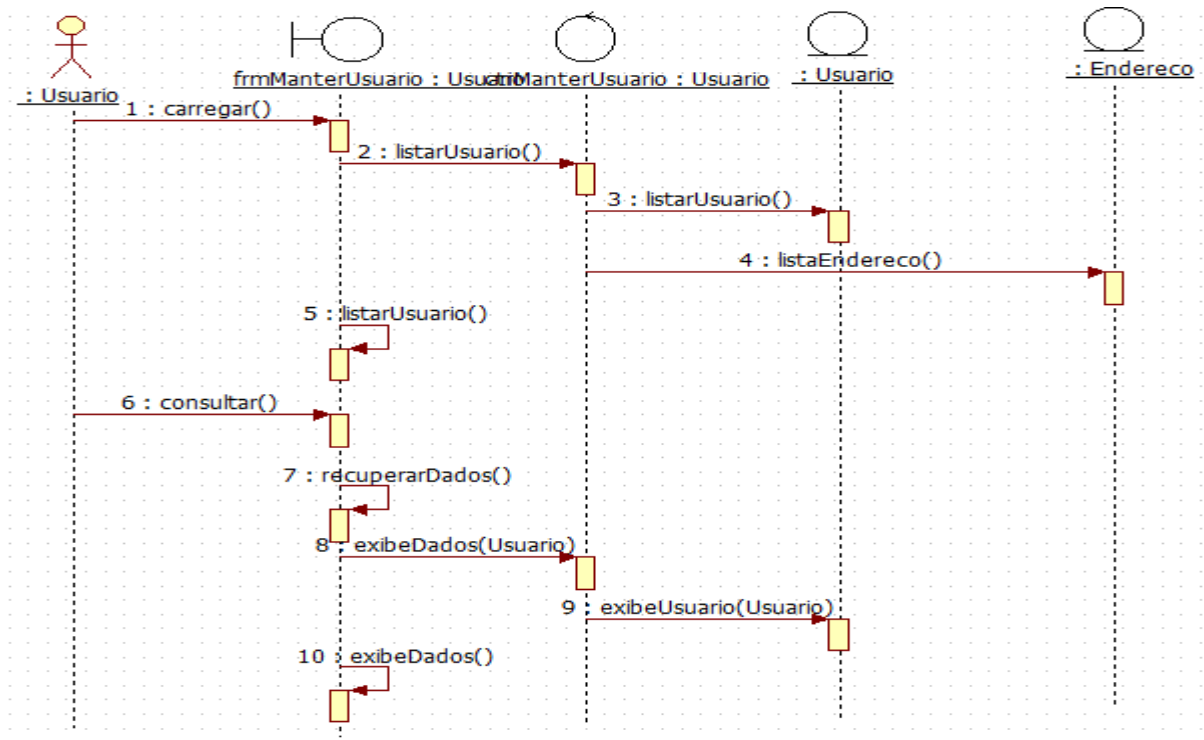


Ilustração A. 8 – Diagrama de sequência consultar usuários

Caso de uso Manter Usuários Administradores.

Nome do caso de uso:

Manter Dados Usuários Administradores

Ator principal:

Usuário Administrador

Ator Secundário:

Não aplicável

Descrição:

Este caso de uso permite a manutenção dos dados (inclusão, exclusão, alteração e consulta) dos usuários administradores.

Pré-Condições:

Dados do endereço mantidos no sistema.

Pós-Condições:

Dados dos usuários administradores mantidos no sistema.

Fluxo Principal

- 1.O usuário administrador solicita ao sistema a manutenção de usuários administradores.
- 2.OSD apresentar ao usuário administrador a tela de manutenção de usuários administradores.
- 3.OSD apresentar ao usuário administrador uma lista de usuários administradores cadastrados, contendo os dados: login, senha, nome, numero, complemento, email, telefone, celular, data do cadastro e renda mensal e ativo. (E1).
- 4.OSD permitir ao usuário administrador selecionar a opção desejada. (Incluir, Alterar, Excluir e Consultar) (A1)
- 5.O usuário administrador seleciona a opção desejada.
- 6.OSD executar o subfluxo correspondente ao tipo de operação recebida (Incluir, Alterar, Excluir, Consultar)

Subfluxos:**Operação Consultar**

7. O usuário administrador seleciona o usuário administrador desejado.
8. O usuário administrador seleciona a opção "Consultar". (A1)
9. O sistema recupera os dados (nome, numero, complemento, email, telefone, celular, data do cadastro, renda mensal e ativo) do usuário administrador desejada (E1).
10. OSD exibir dados do usuário administrador selecionado com os campos preenchidos e desabilitados.

Operação Incluir

7. O usuário administrador seleciona a opção "Incluir".
8. OSD exibir os campos vazios e habilitados.
9. OSD permitir ao usuário administrador informar os dados do usuário: login, senha, nome, numero, complemento, email, telefone, celular, data do cadastro, renda mensal e ativo. (V1, V2, V3, A1)
10. O usuário administrador seleciona a opção "Gravar".
11. OSD gerar o código do usuário administrador.
12. OSD persistir os dados do usuário administrador (E2, RN1).
13. OSD apresentar mensagem ao usuário administrador informando o sucesso da realização da operação.
14. OSD recarregar a lista de usuários administradores cadastrados.

Operação Alterar

7. O usuário administrador seleciona o usuário administrador desejado.
8. O usuário administrador seleciona a opção "Alterar".

9. O sistema recupera os dados (nome, numero, complemento, email, telefone, celular, data do cadastro, renda mensal e ativo) do usuário administrador desejado (E1).
10. OSD exibir dados do usuário administrador selecionado com os campos preenchidos e habilitados. (E1)
11. OSD permitir ao usuário alterar os dados do usuário: senha, nome, numero, complemento, email, telefone, celular, data do cadastro, renda mensal e ativo. (V1, V2, V3)
12. O usuário administrador seleciona a opção “Gravar”.
13. OSD persistir os dados do usuário administrador (E2, RN1).
14. OSD apresentar mensagem ao usuário administrador informando o sucesso da realização da operação.
15. OSD recarregar a lista de usuários administradores cadastrados.

Operação Excluir

7. OSD permitir ao usuário administrador selecionar um usuário administrador. (A1)
8. O usuário administrador seleciona a opção “Excluir”.
9. OSD confirmar a operação.
10. OSD excluir os dados do usuário administrador. (E3)
11. OSD apresentar mensagem ao usuário administrador informando o sucesso da realização da operação.
12. OSD recarregar a lista de usuários administradores cadastrados.

Requisitos não funcionais:

Não Aplicável

Fluxos de Exceção:

E1. O sistema não consegue recuperar os dados de um usuário administrador.

1. OSD exibir mensagem de erro.

E2. O sistema não consegue armazenar/atualizar os dados de um usuário administrador.

1. OSD exibir mensagem de erro.

E3. O sistema não consegue excluir os dados de um usuário administrador.

1. OSD exibir mensagem de erro.

Fluxos Alternativos:

A1. O usuário cancela a operação.

O sistema deve limpar os campos da interface.

O fluxo retorna ao passo quatro do fluxo principal.

Regras de negócio:

RN1: Os usuários administradores têm obrigatoriamente um endereço.

Validações

V1: Os campos login, senha e nome não podem ser vazios.

V2: Os campos número, complemento e email não podem ser vazios.

V3: Os campos telefone, celular, data de cadastro, renda mensal e ativo não podem ser vazios.

Diagrama de Sequência Incluir Usuários Administradores

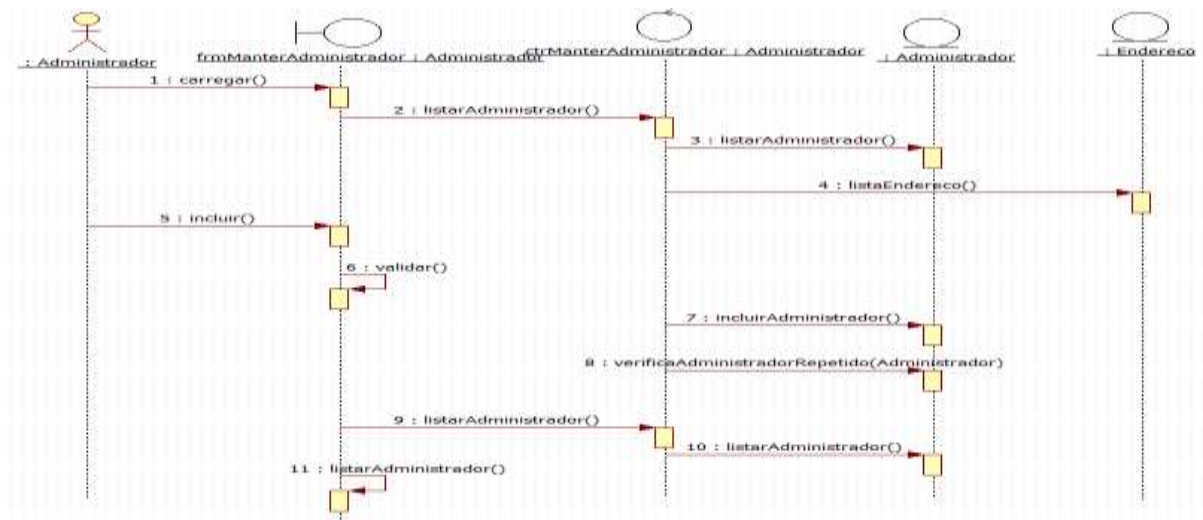


Ilustração A. 9 – Diagrama de sequência incluir usuários Administradores

Diagrama de Sequência Excluir Usuários Administradores

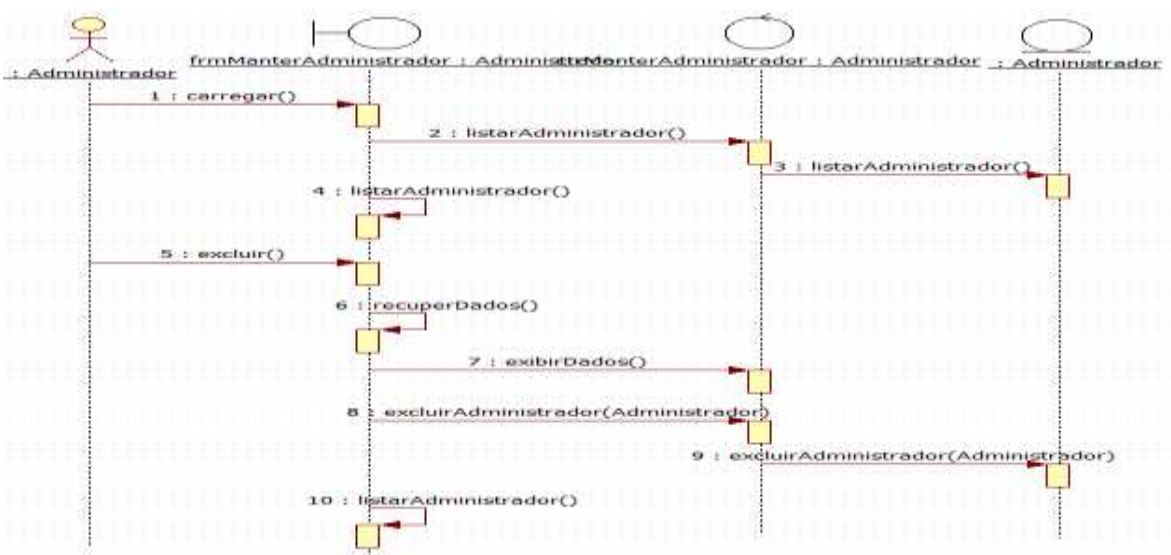


Ilustração A. 10 – Diagrama de sequência excluir usuários Administradores

Diagrama de Sequência Alterar Usuários Administradores

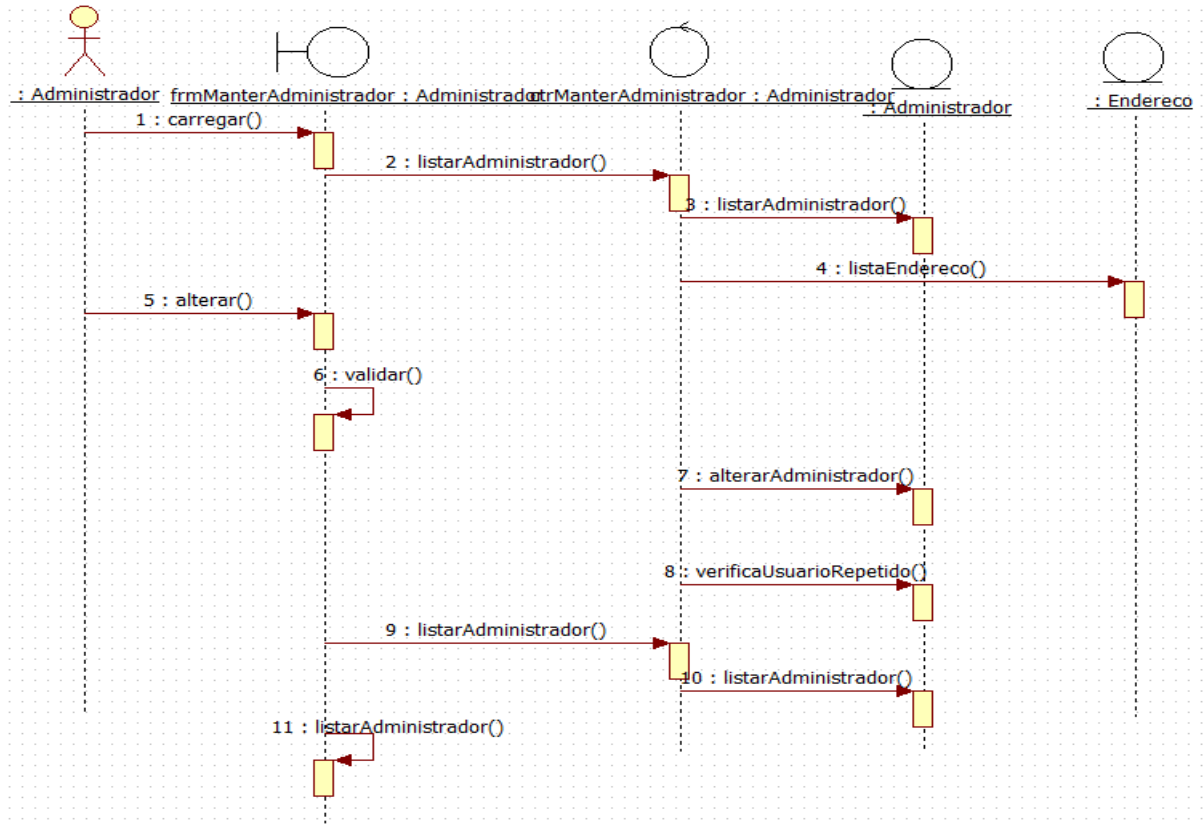


Ilustração A. 11 – Diagrama de sequência alterar usuários Administradores

Diagrama de Sequência Consultar Usuários Administradores

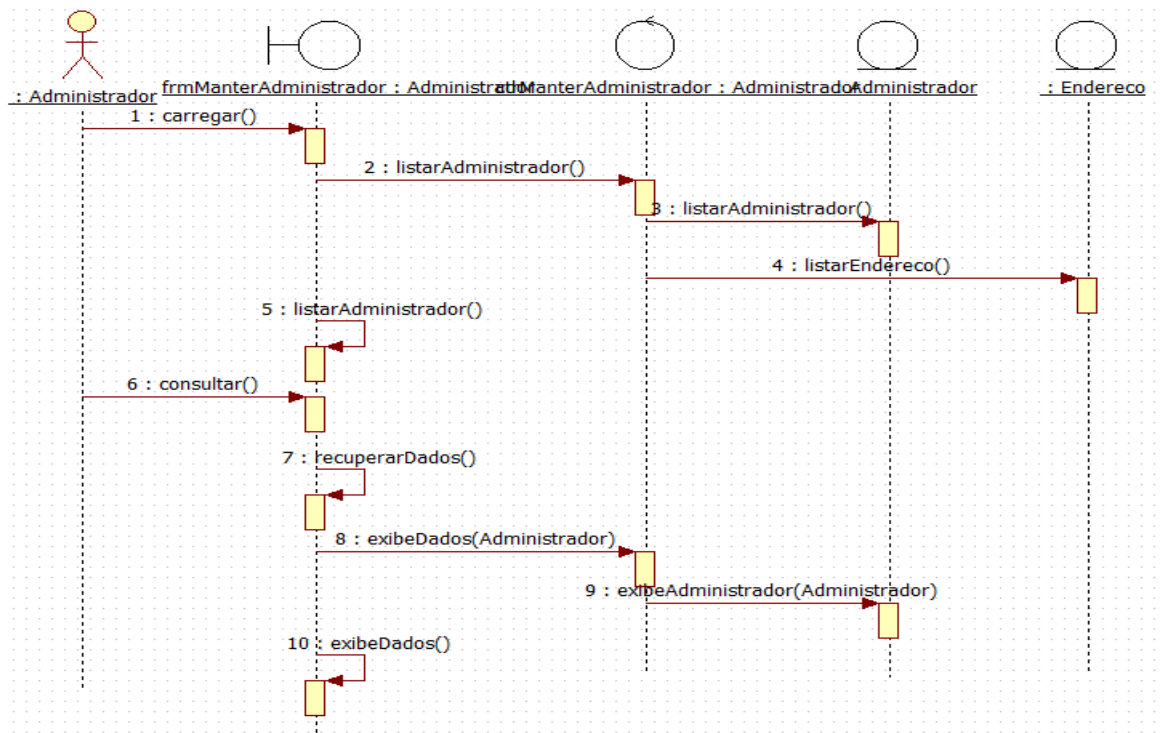


Ilustração A. 12 – Diagrama de sequência consultar usuários Administradores

Caso de uso Manter Usuários Corretores.

Nome do caso de uso:

Manter Dados Usuários Corretores

Ator principal:

Usuário Corretor

Ator Secundário:

Não aplicável

Descrição:

Este caso de uso permite a manutenção dos dados (inclusão, exclusão, alteração e consulta) dos usuários corretores.

Pré-Condições:

Dados do endereço mantidos no sistema.

Pós-Condições:

Dados dos usuários corretores mantidos no sistema.

Fluxo Principal

- 1.O usuário corretor solicita ao sistema a manutenção de usuários corretores.
- 2.OSD apresentar ao usuário corretor a tela de manutenção de usuários corretores.
- 3.OSD apresentar ao usuário corretor uma lista de usuários corretores cadastrados, contendo os dados: login, senha, nome, numero, complemento, email, telefone, celular, data do cadastro e renda mensal, crci, comissão e uma lista de clientes. (E1).
- 4.OSD permitir ao usuário corretor selecionar a opção desejada. (Incluir, Alterar, Excluir e Consultar) (A1)
- 5.O usuário corretor seleciona a opção desejada.
- 6.OSD executar o subfluxo correspondente ao tipo de operação recebida (Incluir, Alterar, Excluir, Consultar)

Subfluxos:

Operação Consultar

7. O usuário corretor seleciona o usuário corretor desejado.
8. O usuário corretor seleciona a opção "Consultar". (A1)

9. O sistema recupera os dados (nome, numero, complemento, email, telefone, celular, data do cadastro, renda mensal crci, comissão e uma lista de clientes) do usuário corretor desejada (E1).
10. OSD exibir dados do usuário corretor selecionado com os campos preenchidos e desabilitados.

Operação Incluir

7. O usuário corretor seleciona a opção "Incluir".
8. OSD exibir os campos vazios e habilitados.
9. OSD permitir ao usuário corretor informar os dados do usuário: login, senha, nome, numero, complemento, email, telefone, celular, data do cadastro, renda mensal crci, comissão e uma lista de clientes. (V1, V2, V3, A1)
10. O usuário corretor seleciona a opção "Gravar".
11. OSD gerar o código do usuário corretor.
12. OSD persistir os dados do usuário corretor (E2, RN1).
13. OSD apresentar mensagem ao usuário corretor informando o sucesso da realização da operação.
14. OSD recarregar a lista de usuários corretores cadastrados.

Operação Alterar

7. O usuário corretor seleciona o usuário corretor desejado.
8. O usuário corretor seleciona a opção "Alterar".
9. O sistema recupera os dados (nome, numero, complemento, email, telefone, celular, data do cadastro, renda mensal crci, comissão e uma lista de clientes) do usuário corretor desejado (E1).
10. OSD exibir dados do usuário corretor selecionado com os campos preenchidos e habilitados. (E1)
11. OSD permitir ao usuário alterar os dados do usuário: senha, nome, numero, complemento, email, telefone, celular, data do cadastro, renda mensal crci, comissão e uma lista de clientes. (V1, V2, V3)
12. O usuário corretor seleciona a opção "Gravar".
13. OSD persistir os dados do usuário corretor (E2, RN1).
14. OSD apresentar mensagem ao usuário corretor informando o sucesso da realização da operação.
15. OSD recarregar a lista de usuários corretores cadastrados.

Operação Excluir

7. OSD permitir ao usuário corretor selecionar um usuário corretor. (A1)
8. O usuário corretor seleciona a opção "Excluir".
9. OSD confirmar a operação.
10. OSD excluir os dados do usuário corretor. (E3)
11. OSD apresentar mensagem ao usuário corretor informando o sucesso da realização da operação.
12. OSD recarregar a lista de usuários corretores cadastrados.

Requisitos não funcionais:

Não Aplicável

Fluxos de Exceção:

- E1. O sistema não consegue recuperar os dados de um usuário corretor.
 1. OSD exibir mensagem de erro.
- E2. O sistema não consegue armazenar/atualizar os dados de um usuário corretor.
 1. OSD exibir mensagem de erro.
- E3. O sistema não consegue excluir os dados de um usuário corretor.
 1. OSD exibir mensagem de erro.

Fluxos Alternativos:

- A1. O usuário cancela a operação.
 O sistema deve limpar os campos da interface.
 O fluxo retorna ao passo quatro do fluxo principal.

Regras de negócio:

RN1: Os usuários corretores têm obrigatoriamente um endereço.

Validações

- V1: Os campos login, senha e nome não podem ser vazios.
 V2: Os campos número, complemento e email não podem ser vazios.
 V3: Os campos telefone, celular, data de cadastro, renda mensal, crci, comissão e a lista de clientes não podem ser vazios.

Diagrama de Sequência Incluir Usuários Corretores

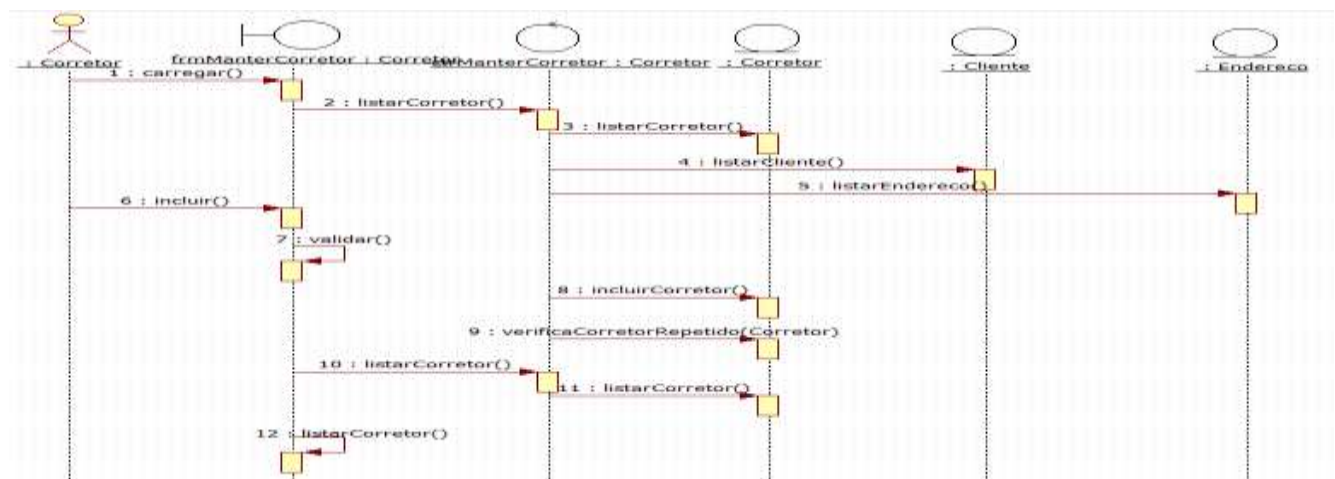


Ilustração A. 13 – Diagrama de sequência incluir usuários Corretores

Diagrama de Sequência Excluir Usuários Corretores

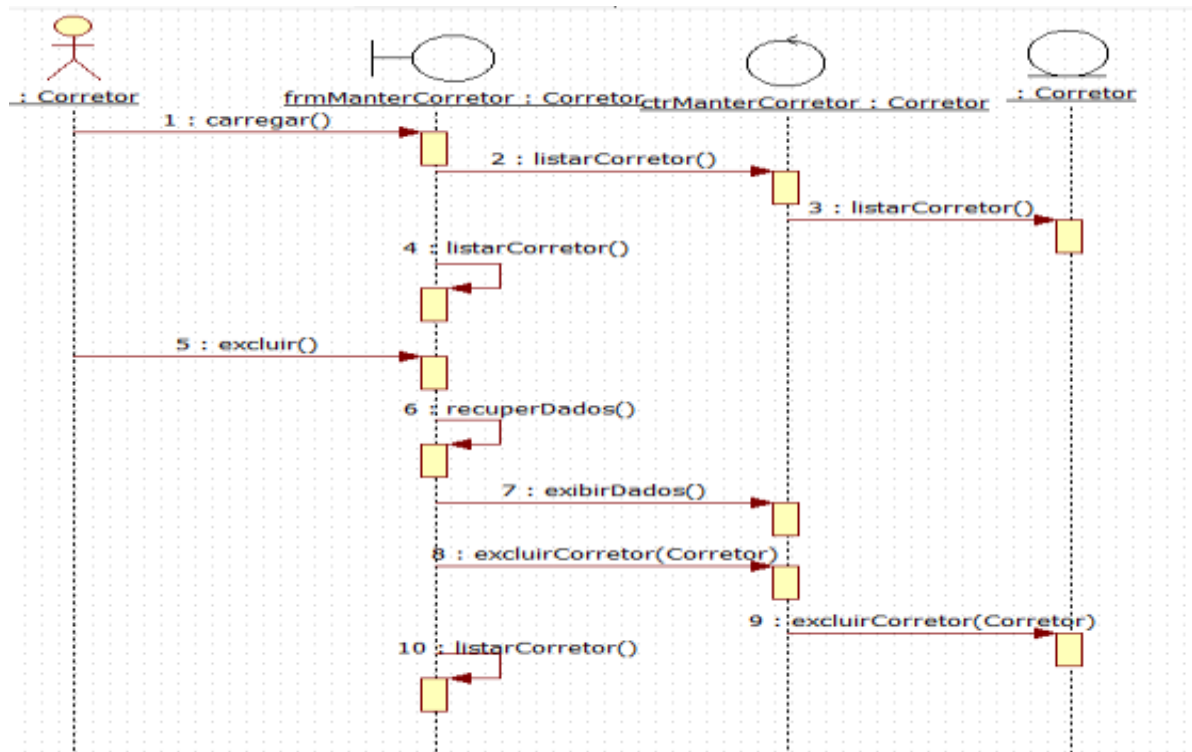


Ilustração A. 14 – Diagrama de sequência excluir usuários Corretores

Diagrama de Sequência Alterar Usuários Corretores

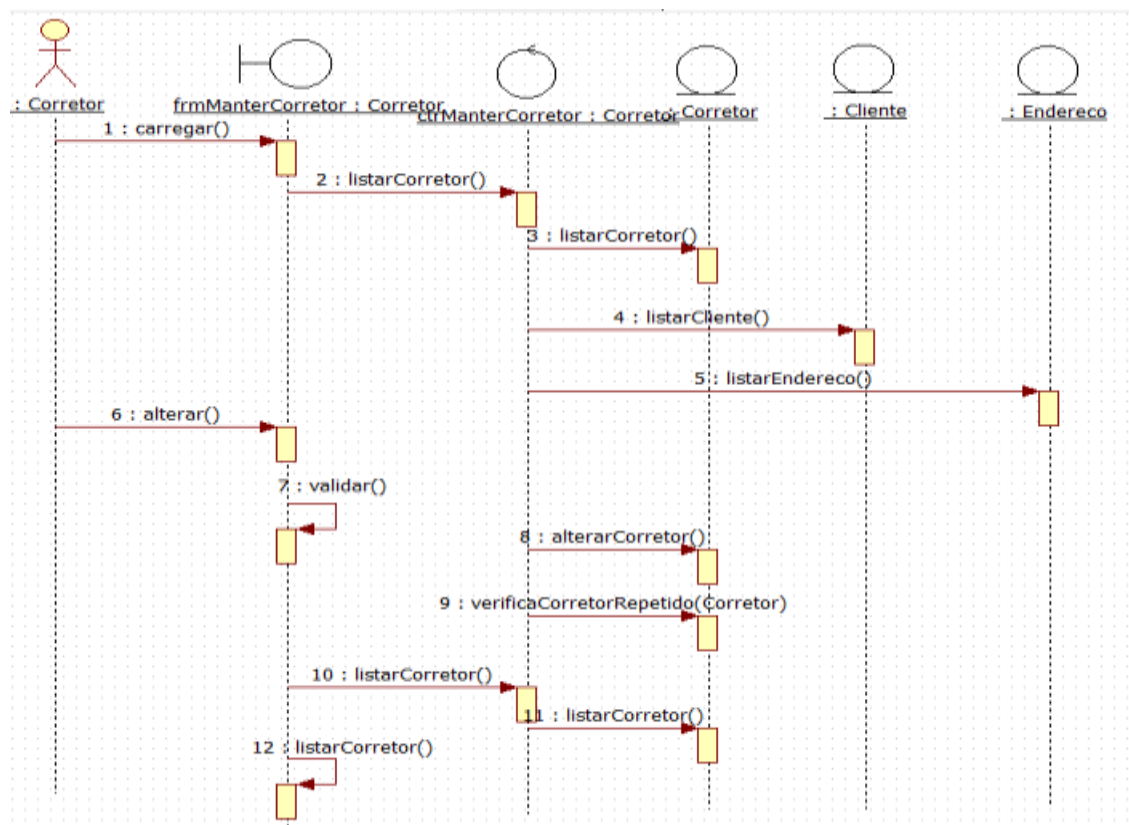


Ilustração A. 15 – Diagrama de sequência alterar usuários Corretores

Diagrama de Sequência Consultar Usuários Corretores

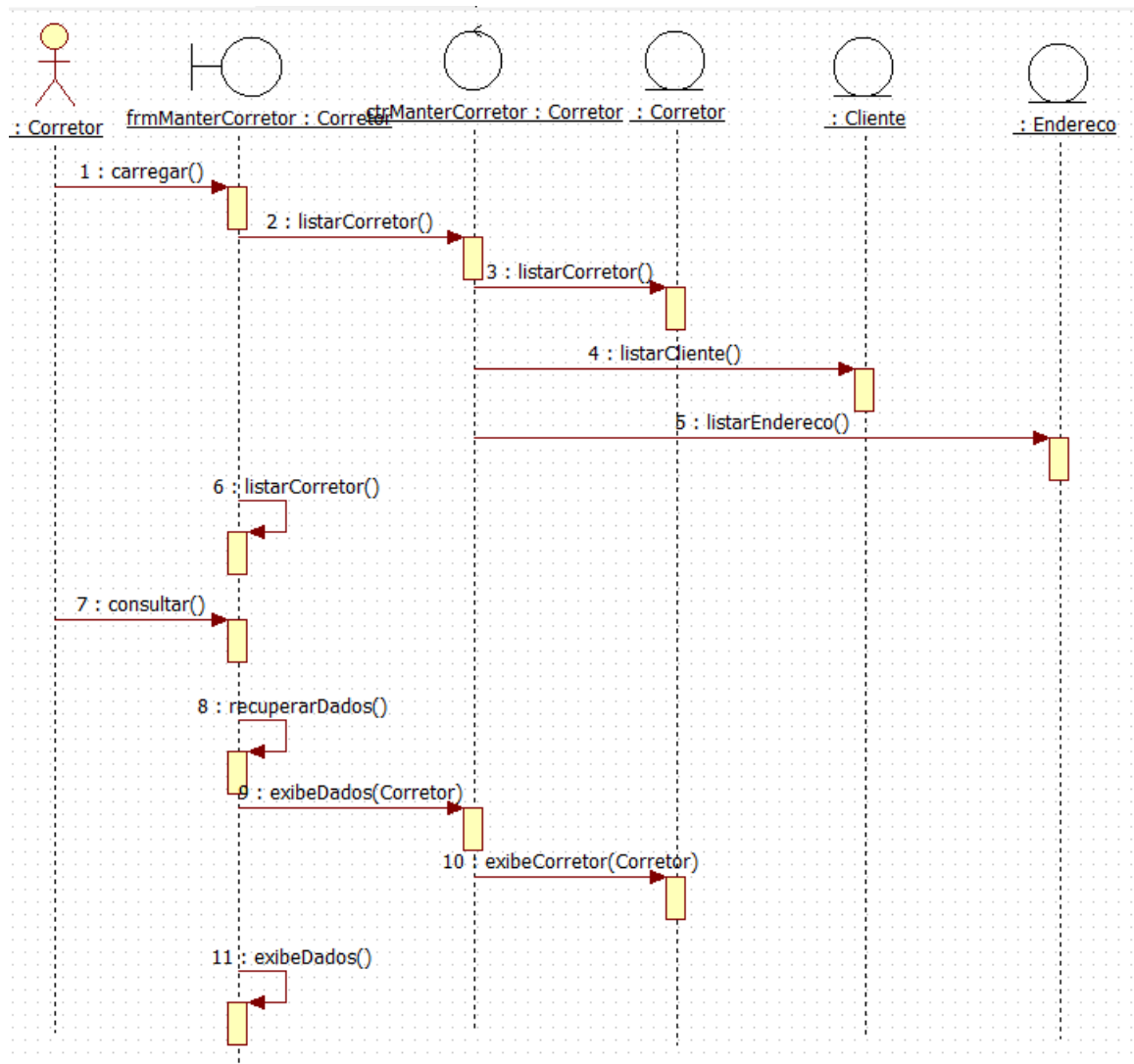


Ilustração A. 16 – Diagrama de sequência consultar usuários Corretores

Caso de uso Manter Usuários Clientes.

Nome do caso de uso:

Manter Dados Usuários Clientes

Ator principal:

Usuário Cliente

Ator Secundário:

Não aplicável

Descrição:

Este caso de uso permite a manutenção dos dados (inclusão, exclusão, alteração e consulta) dos usuários clientes.

Pré-Condições:

Dados do endereço mantidos no sistema.

Pós-Condições:

Dados dos usuários clientes mantidos no sistema.

Fluxo Principal

- 1.O usuário corretor solicita ao sistema a manutenção de usuários clientes.
- 2.OSD apresentar ao usuário cliente a tela de manutenção de usuários clientes.
- 3.OSD apresentar ao usuário cliente uma lista de usuários clientes cadastrados, contendo os dados: login, senha, nome, numero, complemento, email, telefone, celular, data do cadastro e renda mensal, Status do cliente e um usuário corretor. (E1).
- 4.OSD permitir ao usuário corretor selecionar a opção desejada. (Incluir, Alterar, Excluir e Consultar) (A1)
- 5.O usuário corretor seleciona a opção desejada.
- 6.OSD executar o subfluxo correspondente ao tipo de operação recebida (Incluir, Alterar, Excluir, Consultar)

Subfluxos:

Operação Consultar

7. O usuário cliente seleciona o usuário cliente desejado.
8. O usuário cliente seleciona a opção "Consultar". (A1)
9. O sistema recupera os dados (nome, numero, complemento, email, telefone,

celular, data do cadastro, renda mensal, Status do cliente e um usuário corretor) do usuário cliente desejada (E1).

10. OSD exibir dados do usuário cliente selecionado com os campos preenchidos e desabilitados.

Operação Incluir

7. O usuário cliente seleciona a opção “Incluir”.
8. OSD exibir os campos vazios e habilitados.
9. OSD permitir ao usuário corretor informar os dados do usuário: login, senha, nome, numero, complemento, email, telefone, celular, data do cadastro, renda mensal, Status do cliente e um usuário corretor. (V1, V2, V3, A1)
10. O usuário cliente seleciona a opção “Gravar”.
11. OSD gerar o código do usuário cliente.
12. OSD persistir os dados do usuário cliente (E2, RN1).
13. OSD apresentar mensagem ao usuário cliente informando o sucesso da realização da operação.
14. OSD recarregar a lista de usuários clientes cadastrados.

Operação Alterar

7. O usuário cliente seleciona o usuário cliente desejado.
8. O usuário cliente seleciona a opção “Alterar”.
9. O sistema recupera os dados (nome, numero, complemento, email, telefone, celular, data do cadastro, renda mensal, Status do cliente e um usuário corretor) do usuário cliente desejado (E1).
10. OSD exibir dados do usuário cliente selecionado com os campos preenchidos e habilitados. (E1)
11. OSD permitir ao usuário alterar os dados do usuário: senha, nome, numero, complemento, email, telefone, celular, data do cadastro, renda mensal, Status do cliente e um usuário corretor. (V1, V2, V3)
12. O usuário cliente seleciona a opção “Gravar”.
13. OSD persistir os dados do usuário cliente (E2, RN1).
14. OSD apresentar mensagem ao usuário cliente informando o sucesso da realização da operação.
15. OSD recarregar a lista de usuários clientes cadastrados.

Operação Excluir

7. OSD permitir ao usuário cliente selecionar um usuário cliente. (A1)
8. O usuário cliente seleciona a opção “Excluir”.
9. OSD confirmar a operação.
10. OSD excluir os dados do usuário cliente. (E3)
11. OSD apresentar mensagem ao usuário cliente informando o sucesso da realização da operação.
12. OSD recarregar a lista de usuários clientes cadastrados.

Requisitos não funcionais:

Não Aplicável

Fluxos de Exceção:

E1. O sistema não consegue recuperar os dados de um usuário cliente.

1. OSD exibir mensagem de erro.

E2. O sistema não consegue armazenar/atualizar os dados de um usuário cliente.

1. OSD exibir mensagem de erro.

E3. O sistema não consegue excluir os dados de um usuário cliente.

1. OSD exibir mensagem de erro.

Fluxos Alternativos:

A1. O usuário cancela a operação.

O sistema deve limpar os campos da interface.

O fluxo retorna ao passo quatro do fluxo principal.

Regras de negócio:

RN1: Os usuários clientes têm obrigatoriamente um endereço.

Validações

V1: Os campos login, senha e nome não podem ser vazios.

V2: Os campos número, complemento e email não podem ser vazios.

V3: Os campos telefone, celular, data de cadastro, renda mensal, Status do cliente, não podem ser vazios.

Diagrama de Sequência Incluir Usuários Clientes

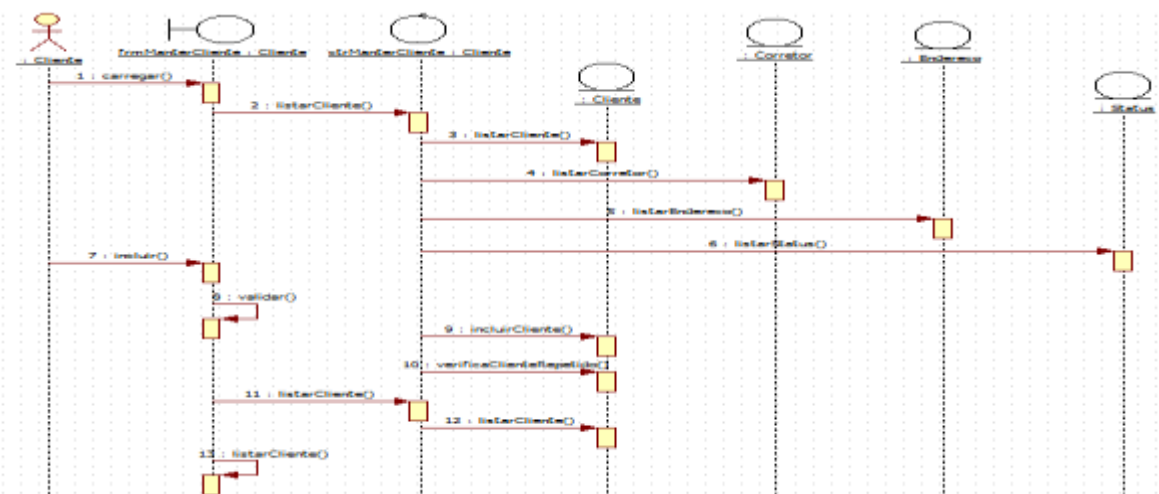


Ilustração A. 17 – Diagrama de sequência incluir usuários Clientes

Diagrama de Sequência Excluir Usuários Clientes

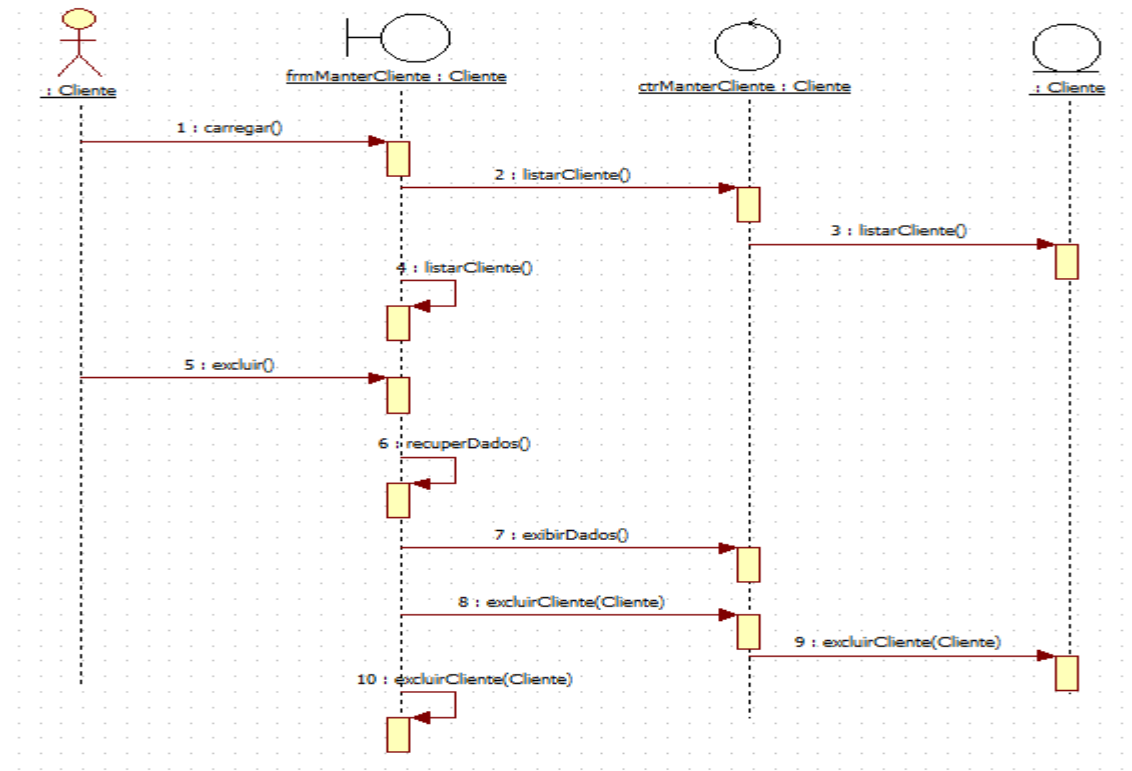


Ilustração A. 18 – Diagrama de sequência excluir usuários Clientes

Diagrama de Sequência Alterar Usuários Clientes

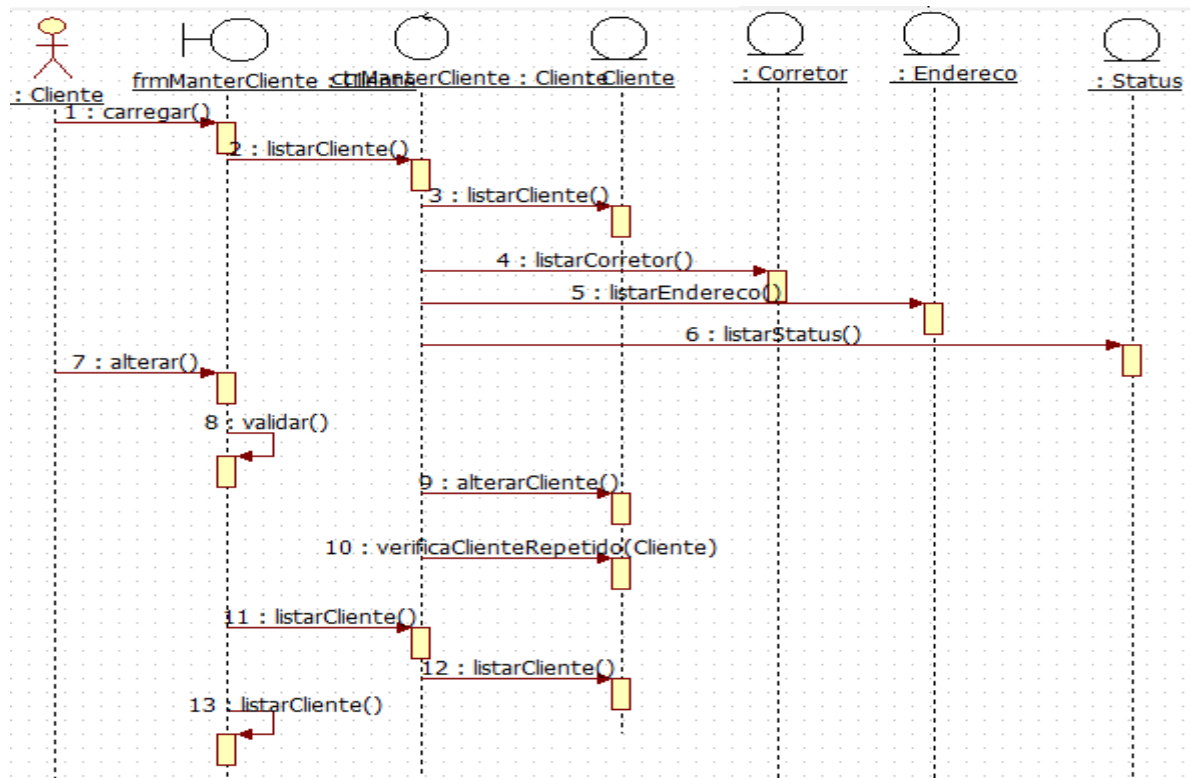


Ilustração A. 19 – Diagrama de sequência alterar usuários Clientes

Diagrama de Sequência Consultar Usuários Clientes

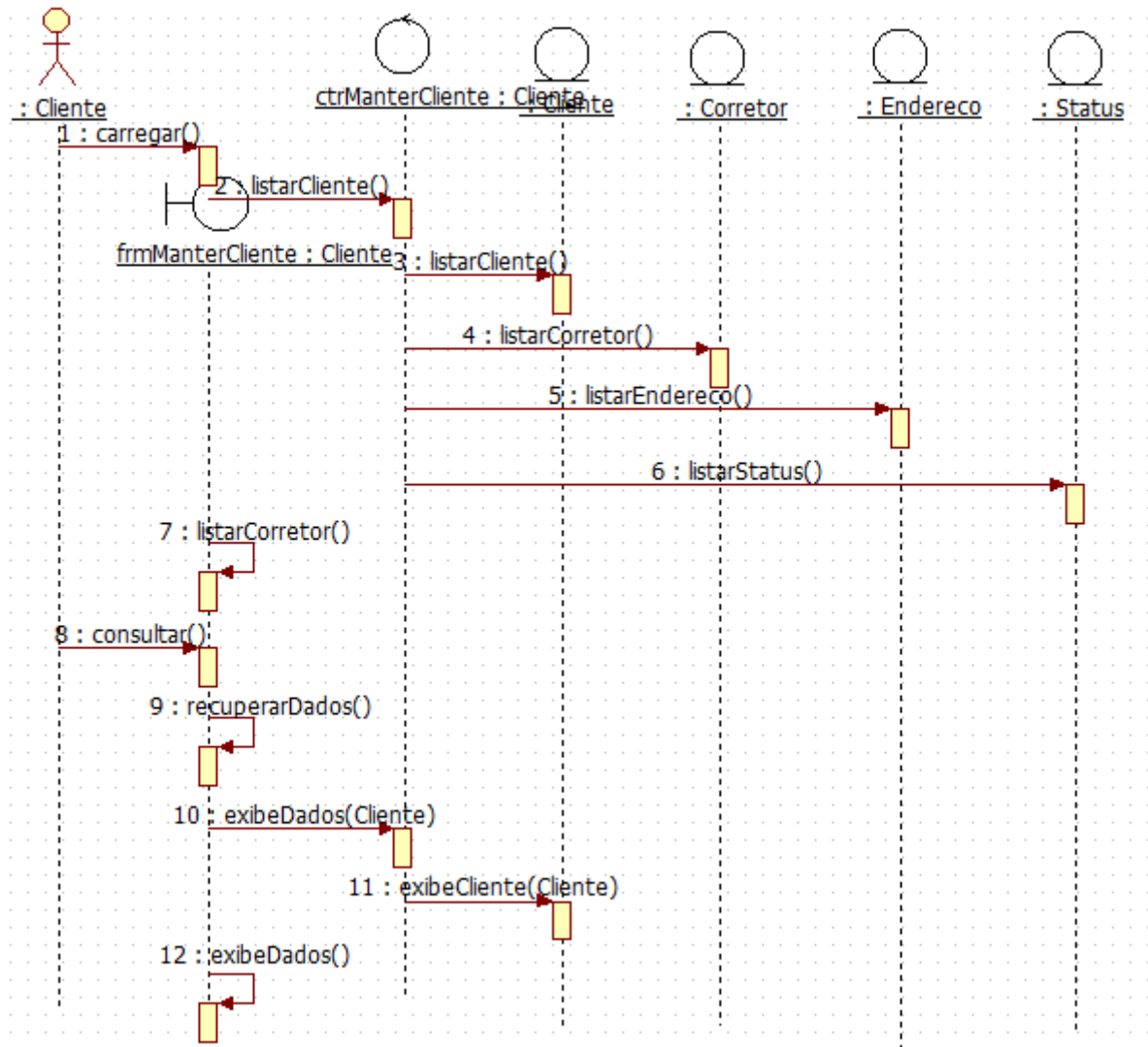


Ilustração A. 20 – Diagrama de sequência consultar usuários Clientes

Caso de uso Manter Status.**Nome do caso de uso:**

Manter Dados Status

Ator principal:

Usuário Administrador

Ator Secundário:

Não aplicável

Descrição:

Este caso de uso permite a manutenção dos dados (inclusão, exclusão, alteração e consulta) dos Status.

Pré-Condições:

Não aplicável.

Pós-Condições:

Dados dos Status mantidos no sistema.

Fluxo Principal

11. O usuário solicita ao sistema a manutenção de Status.
12. OSD apresentar ao usuário a tela de manutenção de Status.
13. OSD apresentar ao usuário uma lista de status cadastrados, contendo os dados: Descrição e Ativo. (E1).
14. OSD permitir ao usuário selecionar a opção desejada. (Incluir, Alterar, Excluir e Consultar) (A1)
15. O usuário seleciona a opção desejada.
16. OSD executar o subfluxo correspondente ao tipo de operação recebida (Incluir, Alterar, Excluir, Consultar)

Subfluxos:**Operação Consultar**

17. O usuário seleciona o status desejado.
18. O usuário seleciona a opção "Consultar". (A1)
19. O sistema recupera os dados (Descrição e Ativo) do Status

desejado (E1).

20. OSD exibir dados do Status selecionado com os campos preenchidos e desabilitados.

Operação Incluir

15. O usuário seleciona a opção “Incluir”.
16. OSD exibir os campos vazios e habilitados.
17. OSD permitir ao usuário informar os dados do status: Descrição, Ativo selecionando o Status. (V1, V2, V3, A1)
18. O usuário seleciona a opção “Gravar”.
19. OSD gerar o código do status.
20. OSD persistir os dados do status (E2, RN1).
21. OSD apresentar mensagem ao usuário informando o sucesso da realização da operação.
22. OSD recarregar a lista de status cadastrados.

Operação Alterar

16. O usuário seleciona o endereço desejado.
17. O usuário seleciona a opção “Alterar”.
18. O sistema recupera os dados (Descrição e Ativo do status desejado (E1).
19. OSD exibir dados do status selecionado com os campos preenchidos e habilitados. (E1)
20. OSD permitir ao usuário alterar os dados do status: Descrição e Ativo, selecionando o status. (V1, V2, V3)
21. O usuário seleciona a opção “Gravar”.
22. OSD persistir os dados do status (E2, RN1).
23. OSD apresentar mensagem ao usuário informando o sucesso da realização da operação.
24. OSD recarregar a lista de status cadastrados.

Operação Excluir

13. OSD permitir ao usuário selecionar um status. (A1)
14. O usuário seleciona a opção “Excluir”.
15. OSD confirmar a operação.
16. OSD excluir os dados do Status. (E3)
17. OSD apresentar mensagem ao usuário informando o sucesso da realização da operação.
18. OSD recarregar a lista de status cadastrados.

Requisitos não funcionais:

Não Aplicável

Fluxos de Exceção:

E1. O sistema não consegue recuperar os dados de um status.

1. OSD exibir mensagem de erro.

E2. O sistema não consegue armazenar/atualizar os dados de um status.

1. OSD exibir mensagem de erro.

E3. O sistema não consegue excluir os dados de um status.

1. OSD exibir mensagem de erro.

Fluxos Alternativos:

A1. O usuário cancela a operação.

O sistema deve limpar os campos da interface.

O fluxo retorna ao passo quatro do fluxo principal.

Regras de negócio:

RN1: Os Status dos clientes estão relacionados com os usuários clientes.

Validações

V1: O campo Descrição não pode ser vazio.

V2: O campo ativo não pode ser vazio.

Diagrama de Sequência Incluir Status

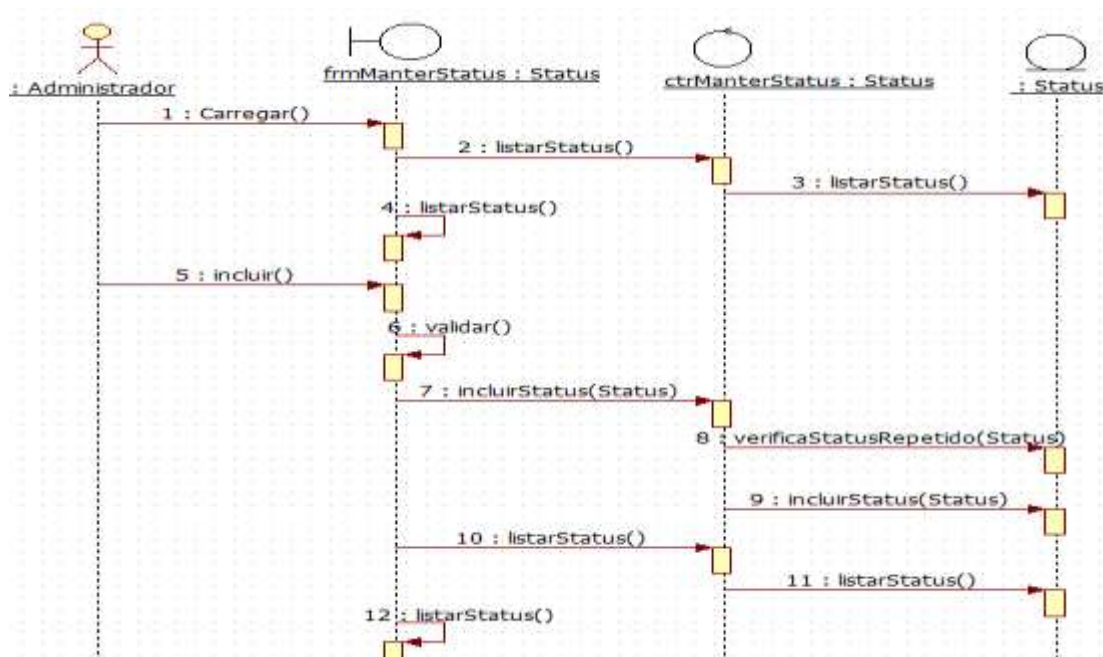


Ilustração A. 21 – Diagrama de sequência incluir status

Diagrama de Sequência Excluir Status

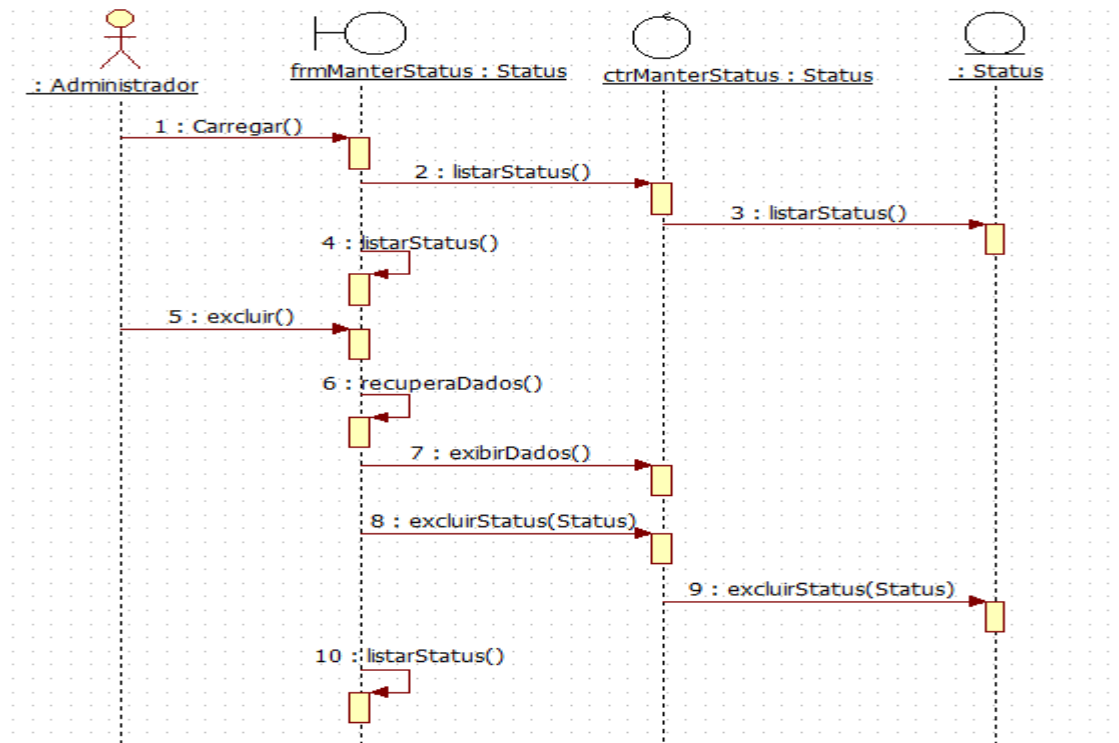


Ilustração A. 22 – Diagrama de sequência excluir status

Diagrama de Sequência Alterar Status

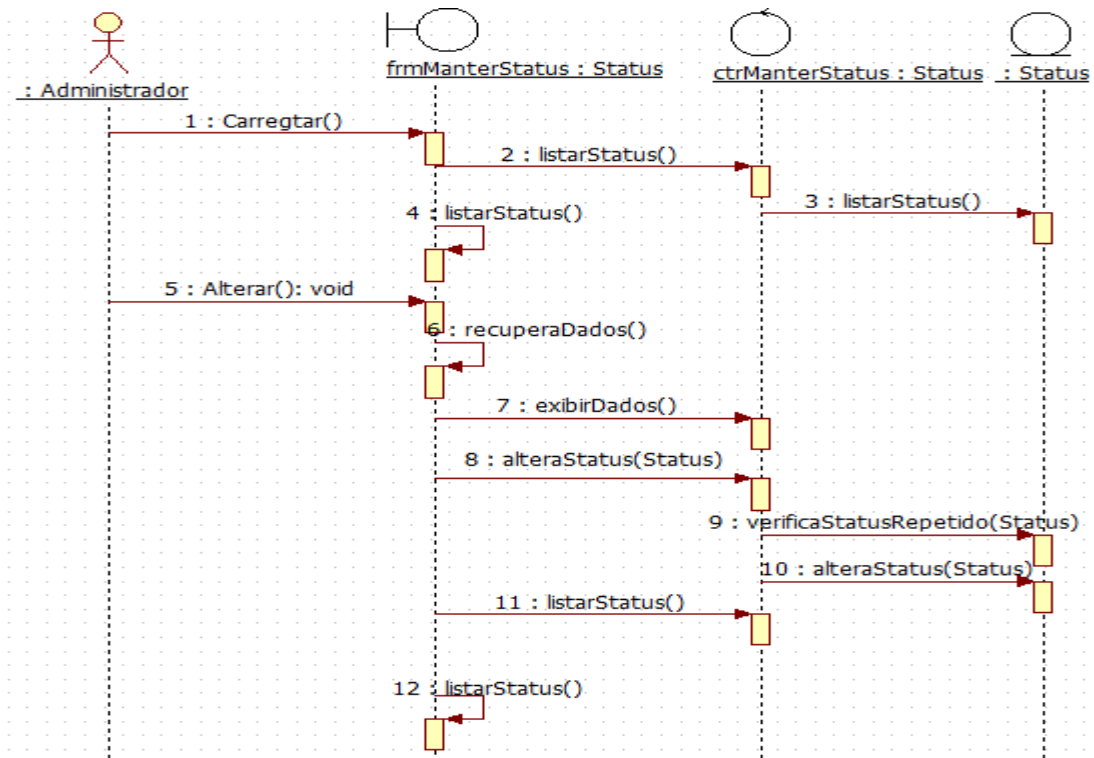


Ilustração A. 23 – Diagrama de sequência alterar Status

Diagrama de Sequência Consultar Status

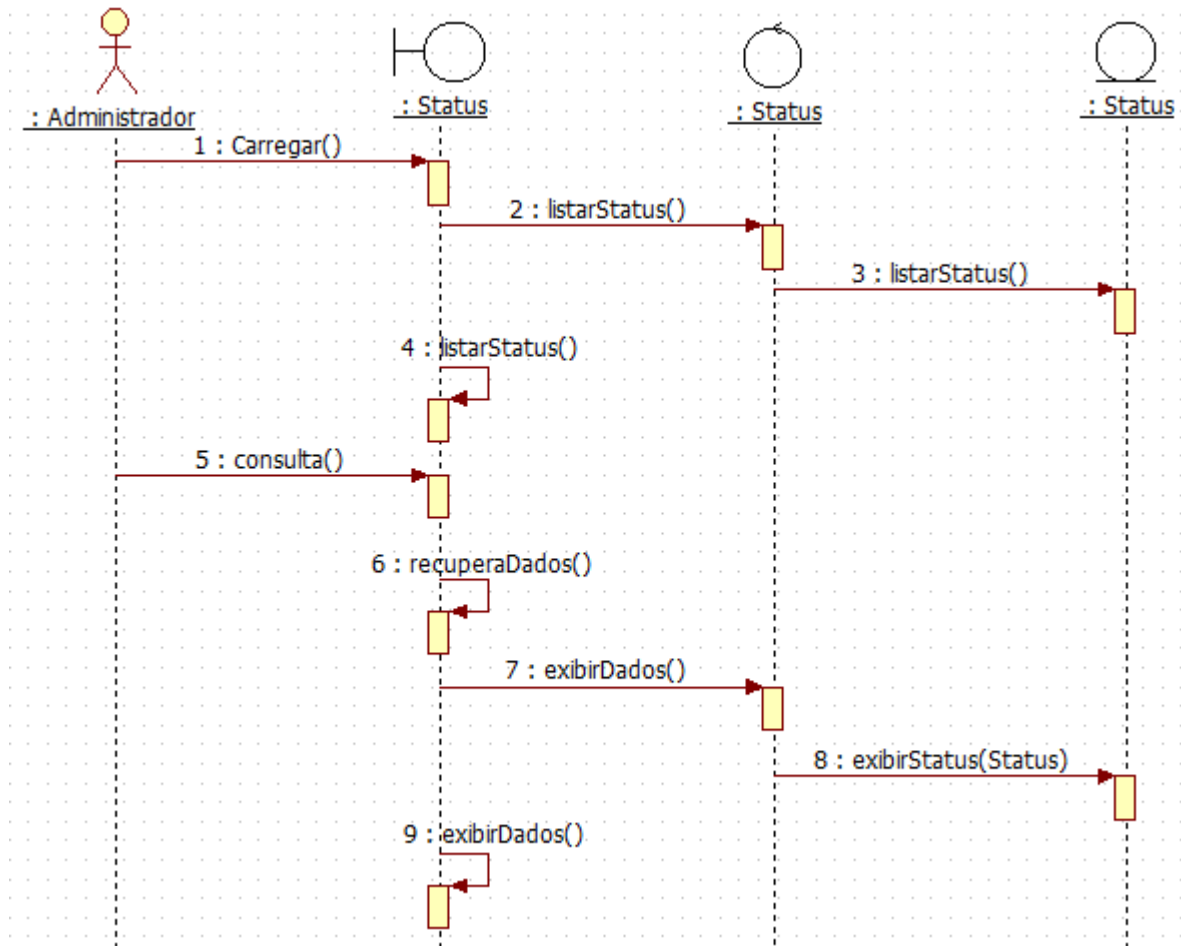


Ilustração A. 24 – Diagrama de sequência consultar status

Caso de uso Manter Tipo Imóvel.

Nome do caso de uso:

Manter Dados Tipo Imóvel

Ator principal:

Usuário Administrador

Ator Secundário:

Não aplicável

Descrição:

Este caso de uso permite a manutenção dos dados (inclusão, exclusão, alteração e

consulta) dos Tipos de Imóveis.

Pré-Condições:

Não aplicável.

Pós-Condições:

Dados dos Tipos de Imóveis mantidos no sistema.

Fluxo Principal

21. O usuário solicita ao sistema a manutenção de Tipos de Imóveis.
22. OSD apresentar ao usuário a tela de manutenção de Tipos de Imóveis.
23. OSD apresentar ao usuário uma lista de Tipos de Imóveis cadastrados, contendo os dados: Descrição. (E1).
24. OSD permitir ao usuário selecionar a opção desejada. (Incluir, Alterar, Excluir e Consultar) (A1)
25. O usuário seleciona a opção desejada.
26. OSD executar o subfluxo correspondente ao tipo de operação recebida (Incluir, Alterar, Excluir, Consultar)

Subfluxos:

Operação Consultar

27. O usuário seleciona o tipo de Imóvel desejado.
28. O usuário seleciona a opção "Consultar". (A1)
29. O sistema recupera os dados (Descrição) do Tipo de Imóvel desejado (E1).
30. OSD exibir dados do Tipo de Imóvel selecionado com os campos preenchidos e desabilitados.

Operação Incluir

23. O usuário seleciona a opção "Incluir".
24. OSD exibir os campos vazios e habilitados.
25. OSD permitir ao usuário informar os dados do Tipo de Imóvel: Descrição selecionando o Tipo de Imóvel. (V1, V2, V3, A1)
26. O usuário seleciona a opção "Gravar".
27. OSD gerar o código do Tipo de Imóvel.
28. OSD persistir os dados do Tipo de Imóvel (E2, RN1).
29. OSD apresentar mensagem ao usuário informando o sucesso da realização da operação.
30. OSD recarregar a lista de Tipos de Imóveis cadastrados.

Operação Alterar

25. O usuário seleciona o Tipo de Imóvel desejado.
26. O usuário seleciona a opção "Alterar".
27. O sistema recupera os dados (Descrição do Tipo de Imóvel desejado (E1).
28. OSD exibir dados do Tipo de Imóvel selecionado com os campos preenchidos e habilitados. (E1)
29. OSD permitir ao usuário alterar os dados do Tipo de Imóvel: Descrição, selecionando o Tipo de Imóvel. (V1, V2, V3)
30. O usuário seleciona a opção "Gravar".
31. OSD persistir os dados do Tipo de Imóvel (E2, RN1).
32. OSD apresentar mensagem ao usuário informando o sucesso da realização da operação.
33. OSD recarregar a lista de Tipos de Imóveis cadastrados.

Operação Excluir

19. OSD permitir ao usuário selecionar um Tipo de Imóvel. (A1)
20. O usuário seleciona a opção "Excluir".
21. OSD confirmar a operação.
22. OSD excluir os dados do Tipo de Imóvel. (E3)
23. OSD apresentar mensagem ao usuário informando o sucesso da realização da operação.
24. OSD recarregar a lista de Tipos de Imóveis cadastrados.

Requisitos não funcionais:

Não Aplicável

Fluxos de Exceção:

E1. O sistema não consegue recuperar os dados de um Tipo de Imóvel.

1. OSD exibir mensagem de erro.

E2. O sistema não consegue armazenar/atualizar os dados de um Tipo de Imóvel.

1. OSD exibir mensagem de erro.

E3. O sistema não consegue excluir os dados de um Tipo de Imóvel.

1. OSD exibir mensagem de erro.

Fluxos Alternativos:

A1. O usuário cancela a operação.

O sistema deve limpar os campos da interface.

O fluxo retorna ao passo quatro do fluxo principal.

Regras de negócio:

RN1: Os Tipos de Imóveis estão relacionados com os imóveis.

Validações

V1: O campo Descrição não pode ser vazio.

V2: O campo Descrição não pode ser repetido.

Diagrama de Sequência Incluir Tipo de Imóvel.

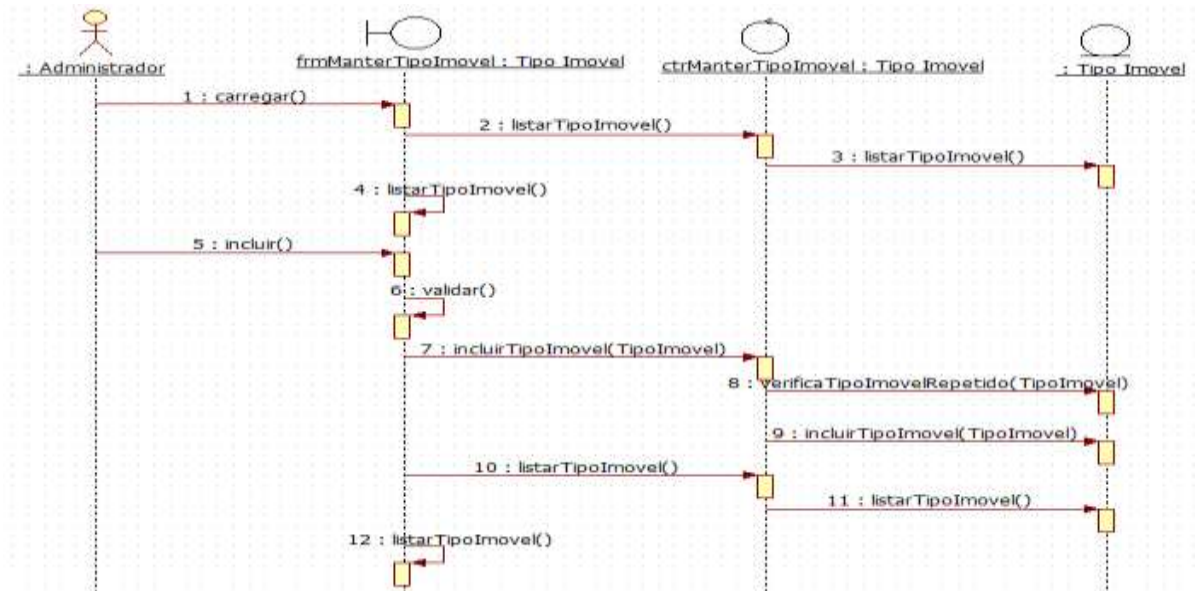


Ilustração A. 25 – Diagrama de sequência Tipo de Imóvel.

Diagrama de Sequência Excluir Tipo de Imóvel.

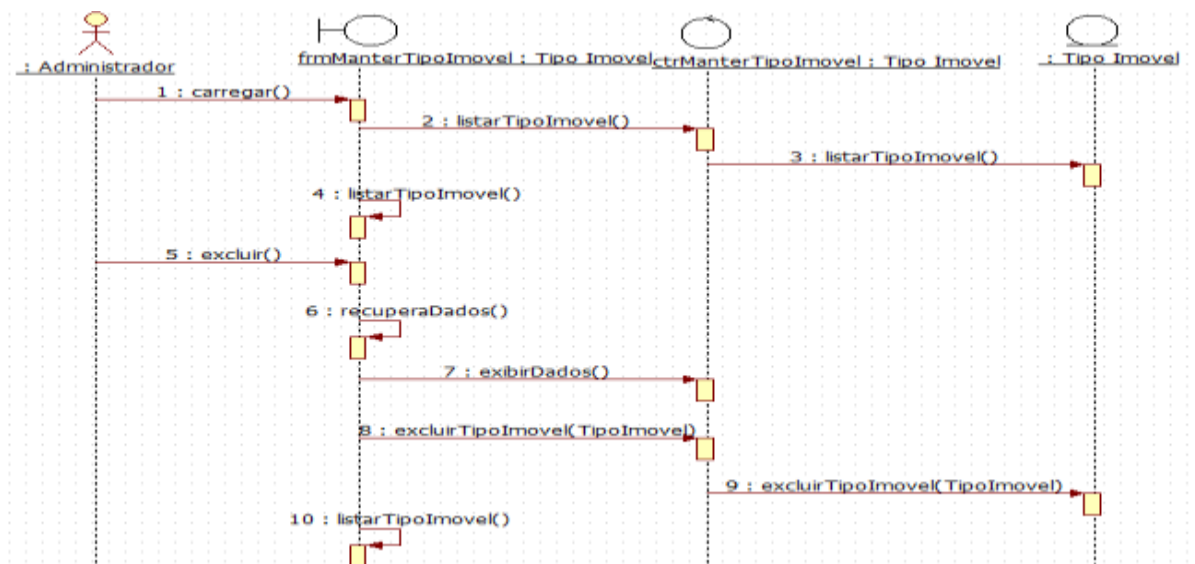


Ilustração A. 26 – Diagrama de sequência excluir Tipo de Imóvel.

Diagrama de Sequência Alterar Tipo de Imóvel.

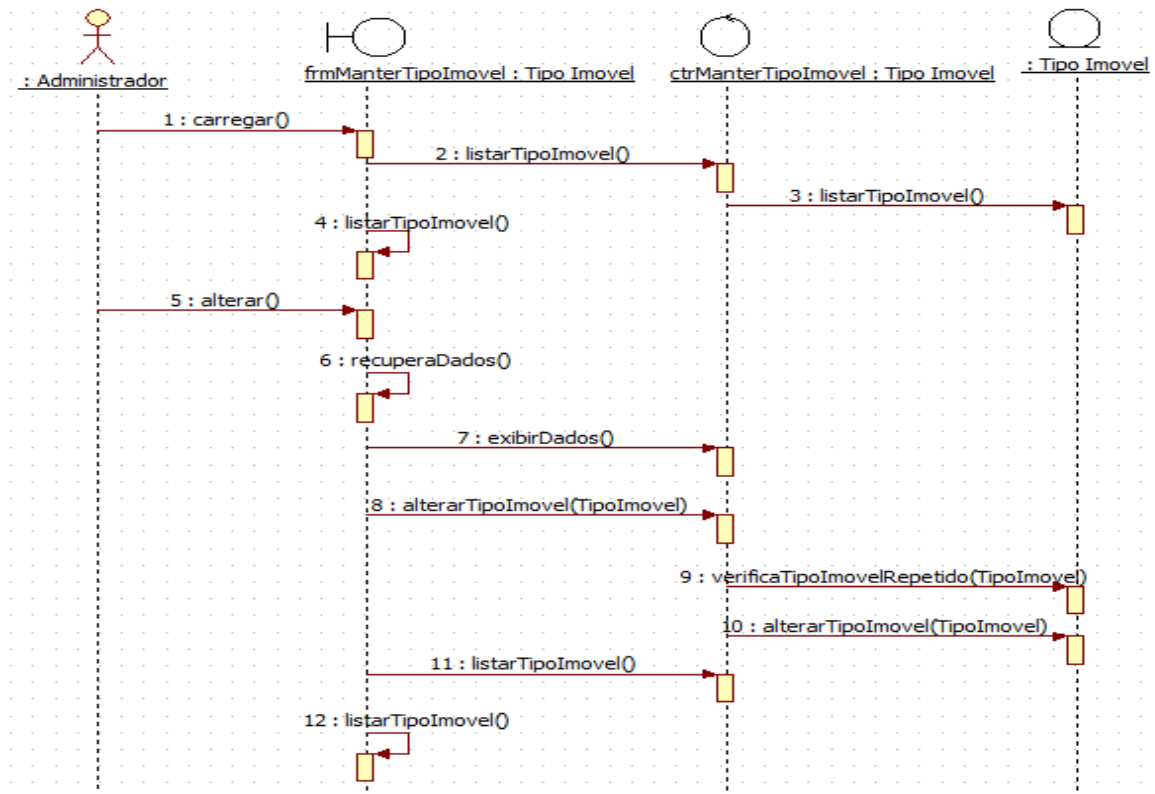


Ilustração A. 27 – Diagrama de sequência alterar Tipos de Imóveis

Diagrama de Sequência Consultar Tipo de Imóvel

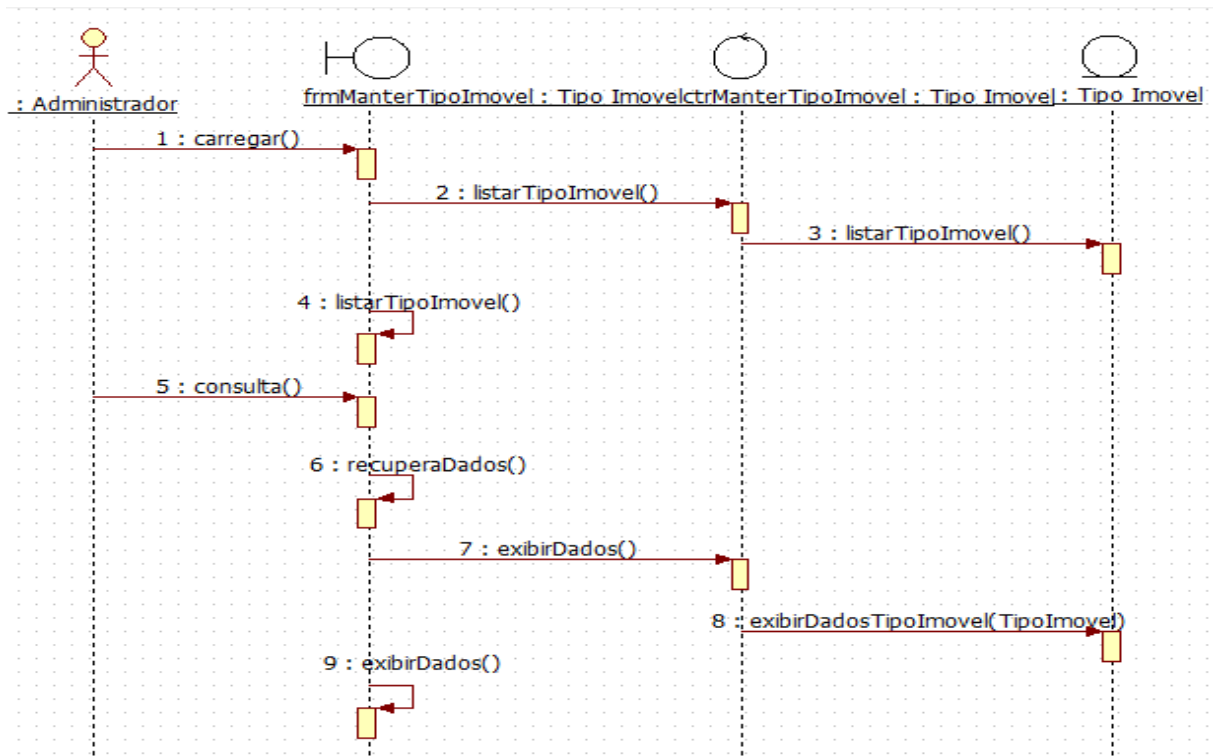


Ilustração A. 28 – Diagrama de sequência consultar Tipo de Imóvel

Caso de uso Manter Agenda de Imóveis.

Nome do caso de uso:

Manter Dados Agendar Imóveis

Ator principal:

Usuário Corretor ou Usuário Cliente

Ator Secundário:

Não aplicável

Descrição:

Este caso de uso permite a manutenção dos dados (inclusão, exclusão e consulta) da Agenda de Imóveis.

Pré-Condições:

Dados dos imóveis, clientes e corretores mantidos no sistema.

Pós-Condições:

Dados dos agendamentos de imóveis mantidos no sistema.

Fluxo Principal

- 1.O usuário solicita ao sistema a manutenção de agendamento de imóveis.
- 2.OSD apresentar ao usuário a tela de manutenção de agendamento de imóveis.
- 3.OSD apresentar ao usuário uma lista de agendamento de imóveis cadastrados, contendo os dados: Imóvel, cliente, corretor, Data do agendamento, hora início agendamento e hora fim agendamento (E1).
- 4.OSD permitir ao usuário selecionar a opção desejada. (Incluir, Alterar, Excluir e Consultar) (A1)
- 5.O usuário seleciona a opção desejada.
- 6.OSD executar o subfluxo correspondente ao tipo de operação recebido (Incluir, Alterar, Excluir, Consultar)

Subfluxos:

Operação Consultar

- 7.O usuário seleciona o agendamento do imóvel desejado.
- 8.O usuário seleciona a opção "Consultar". (A1)
- 9.O sistema recupera os dados (Imóvel, cliente, corretor, Data do agendamento,

hora início agendamento e hora fim agendamento) do agendamento de imóveis desejado (E1).

10.OSD exibir dados do pedido selecionado com os campos preenchidos e desabilitados.

Operação Incluir

7.O usuário seleciona a opção “Incluir”.

8.OSD exibir os campos vazios e habilitados.

9.OSD permitir ao usuário informar os dados do agendamento de imóveis: Imóvel, cliente, corretor, Data do agendamento, hora início agendamento e hora fim agendamento. (V1, V2, A1)

10.O usuário seleciona a opção “Gravar”.

11.OSD gerar o código do agendamento de imóveis.

12.OSD persistir os dados do agendamento de imóveis (E2, RN1).

13.OSD apresentar mensagem ao usuário informando o sucesso da realização da operação.

14.OSD recarregar a lista do agendamento de imóveis cadastrados.

Operação alterar

7.O usuário seleciona a opção “alterar”.

8.OSD O sistema recupera os dados: Imóvel, cliente, corretor, Data do agendamento, hora início agendamento e hora fim agendamento. (V1, V2, A1)

9.O usuário seleciona a opção “Gravar”.

10.OSD gerar o código do agendamento de imóveis.

11.OSD persistir os dados de agendamento de imóveis (E2, RN1).

12.OSD apresentar mensagem ao usuário informando o sucesso da realização da operação.

13.OSD recarregar a lista da agendamento de imóveis cadastrados.

Operação Excluir

7.OSD permitir ao usuário selecionar o agendamento de imóveis. (A1)

8.O usuário seleciona a opção “Excluir”.

9.OSD confirmar a operação.

10.OSD excluir os dados do agendamento de imóveis. (E3)

12.OSD apresentar mensagem ao usuário informando o sucesso da realização da operação.

13.OSD recarregar a lista da agendamento de imóveis cadastrados.

Requisitos não funcionais:

Não Aplicável

Fluxos de Exceção:

E1. O sistema não consegue recuperar os dados do agendamento de imóveis.

1. OSD exibir mensagem de erro.

E2. O sistema não consegue armazenar/atualizar os dados do agendamento de imóveis.

1. OSD exibir mensagem de erro.

E3. O sistema não consegue excluir os dados do agendamento de imóveis.

1. OSD exibir mensagem de erro.

Fluxos Alternativos:

A1. O usuário cancela a operação.

O sistema deve limpar os campos da interface.

O fluxo retorna ao passo quatro do fluxo principal.

Regras de negócio:

RN1: Os agendamentos de imóveis têm obrigatoriamente imóvel, corretor e cliente.

Validações

V1: O campo data do agendamento não pode ser vazio.

V2: a hora inicial do agendamento e a hora final do agendamento não pode ser vazio.

Diagrama de Sequência Incluir Agendamento de Imóveis.

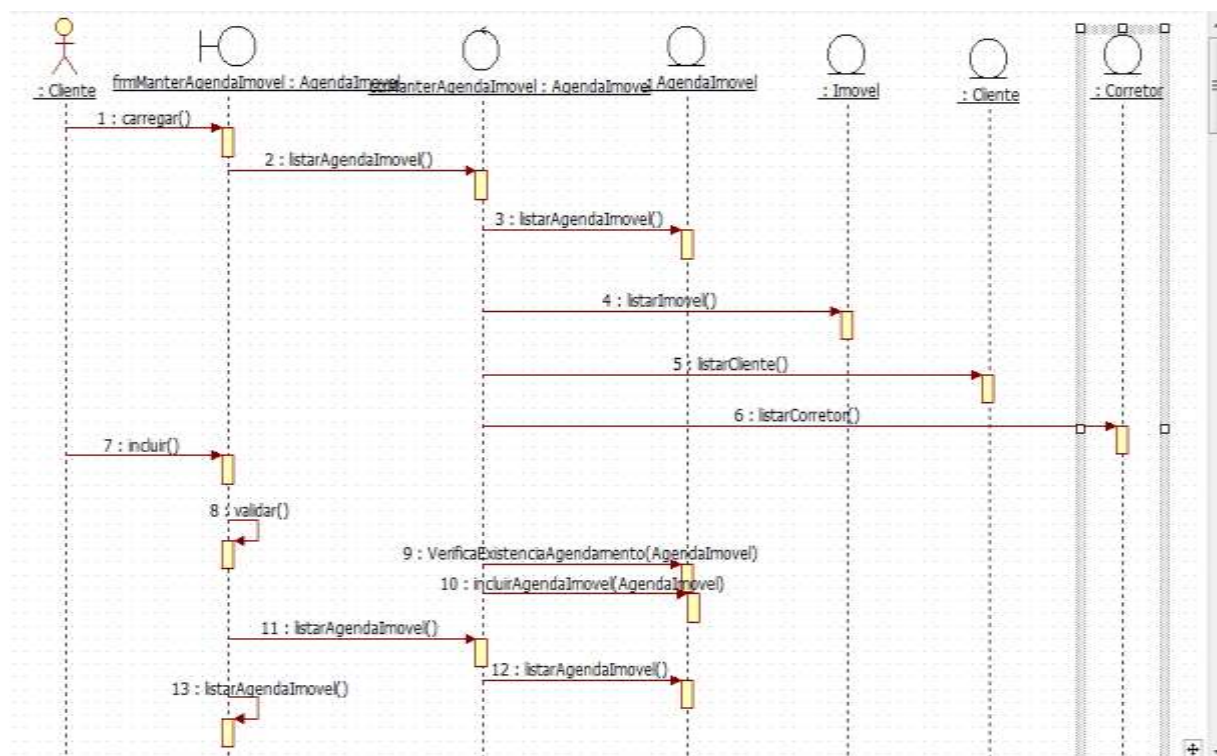


Ilustração A. 29 – Diagrama de sequência incluir Agendamento de Imóveis

Diagrama de Sequência Excluir Agendamento de Imóveis

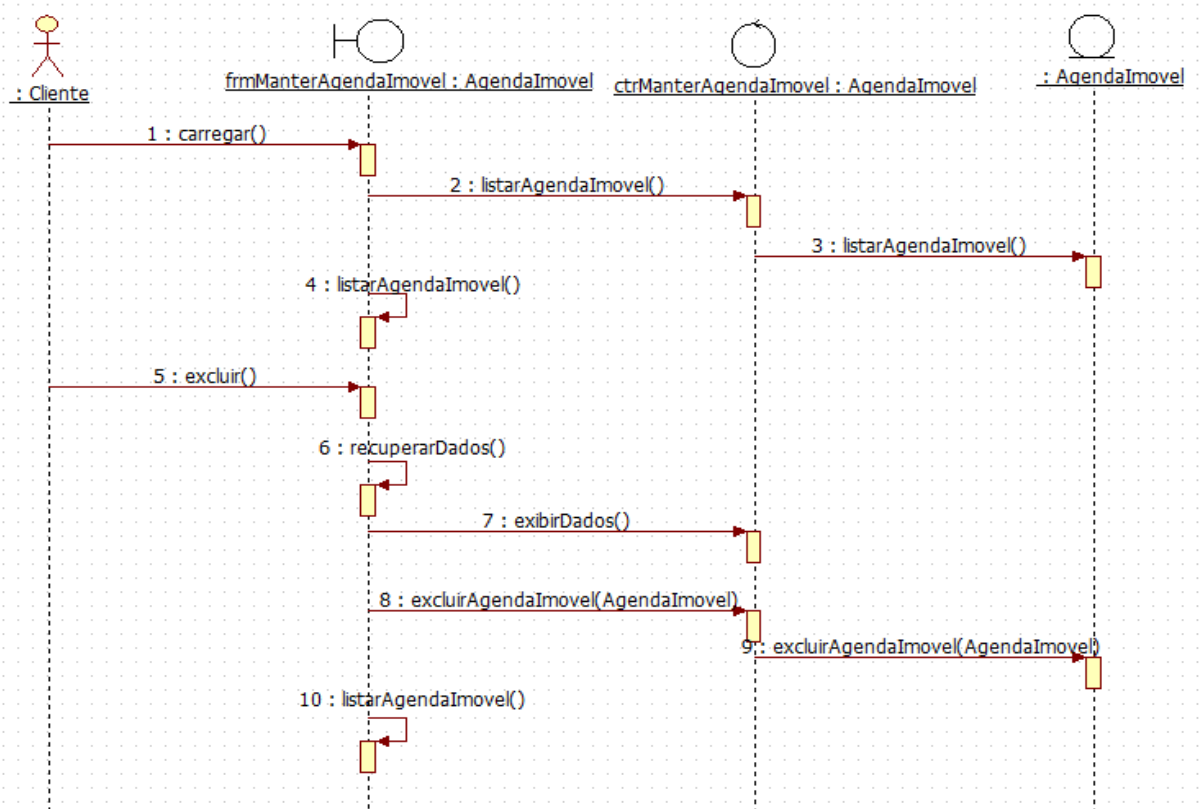


Ilustração A. 30 – Diagrama de sequência excluir agendamento de imóveis

Diagrama de Sequência Alterar Agendamento de Imóveis

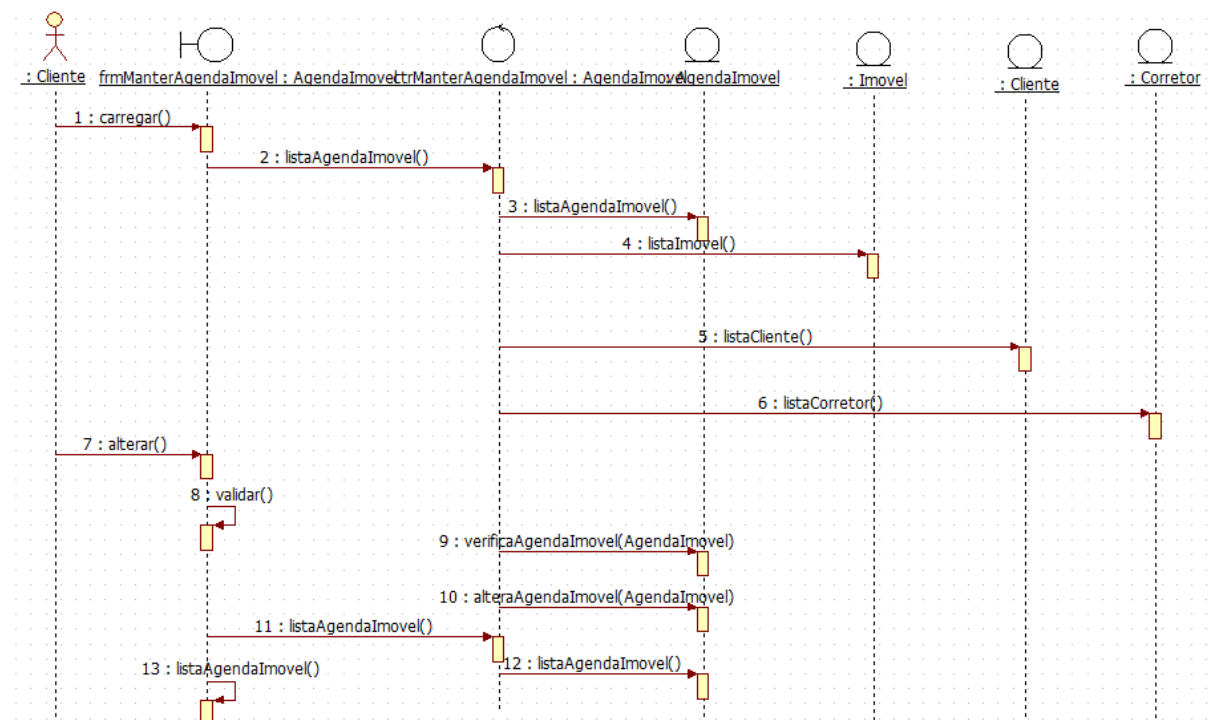


Ilustração A. 31 – Diagrama de sequência alterar agendamento de imóveis

Diagrama de Sequência Consultar Agendamento de Imóveis

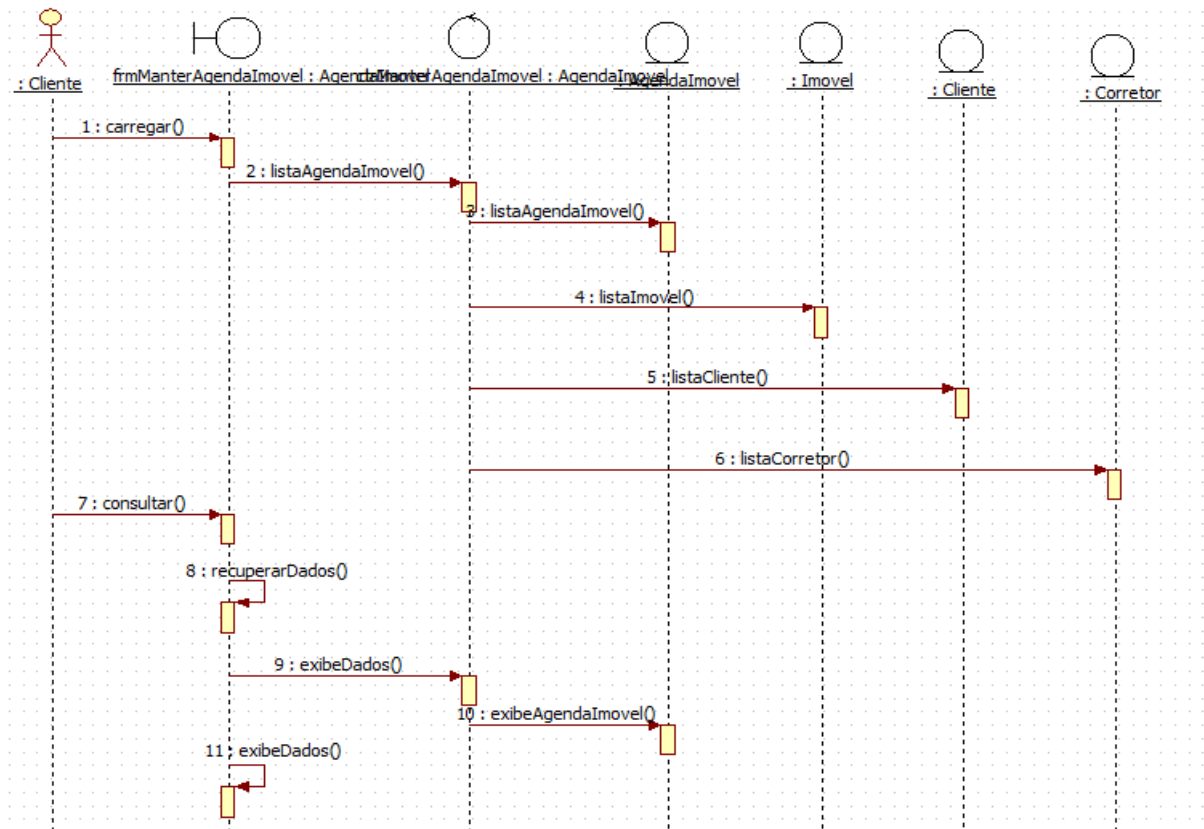


Ilustração A. 32 – Diagrama de sequência consultar Agendamento de Imóveis

Diagrama de Tabelas Relacionais (DTR)

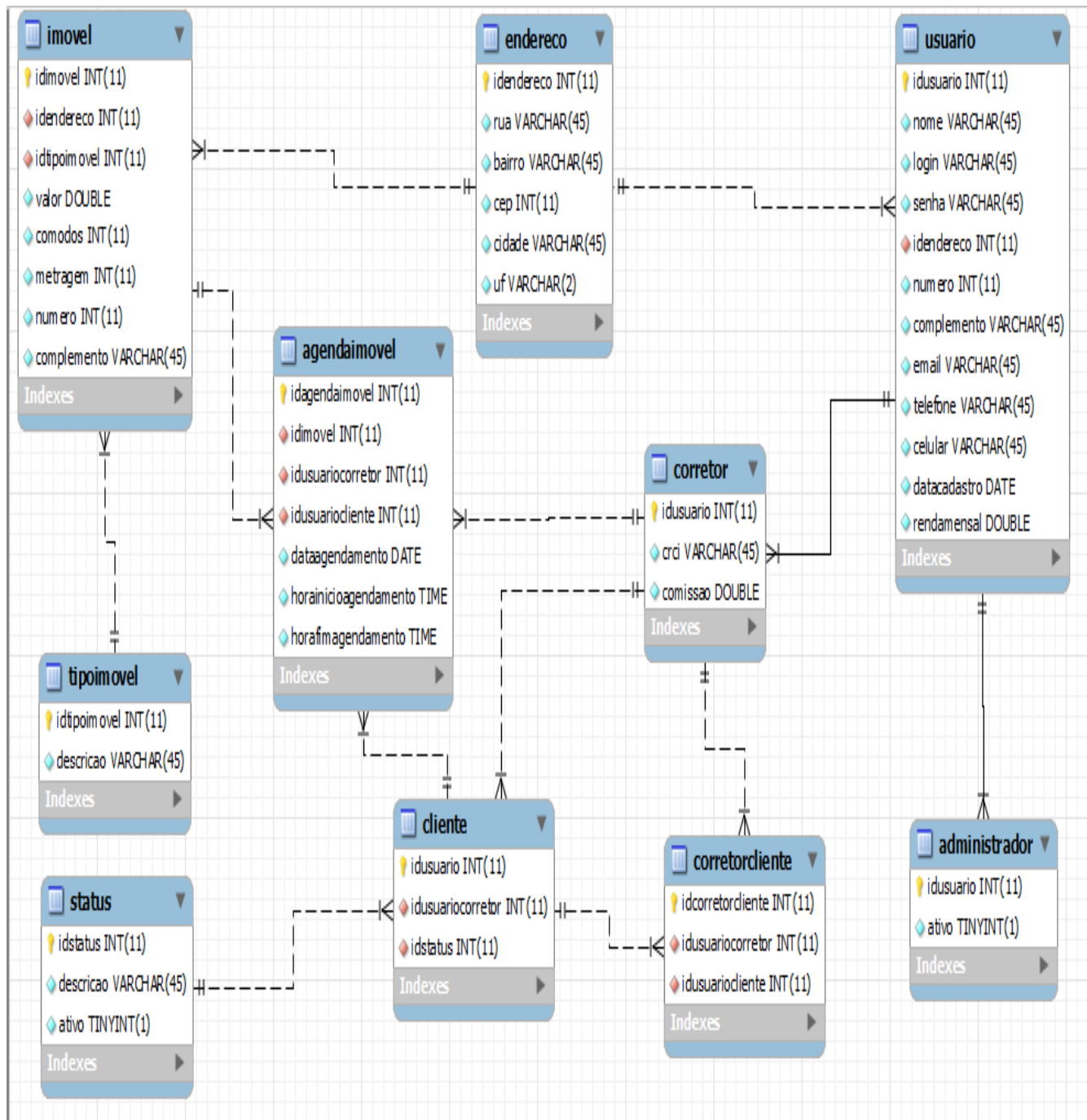


Ilustração A. 33 - Diagrama de tabelas relacionais (DTR) do SISIMOVEIS

Apêndice B – Principais telas do SISIMÓVEIS



Ilustração B. 1 – Tela principal do SISIMOVEIS

SISTEMA DE AGENDAMENTO IMOBILIARIO

Principal Logar

Cadastramento

Endereço

Cliente

Corretor

Imóvel

Tipo de Imóvel

Status

Cadastramento

Agenda Imóvel

Cadastramento

Agenda Imóvel



Rua	Bairro	CEP	Cidade	UF	Apagar	Alterar
Batista de Oliveira 45	Centro	36010360	Juiz de Fora	MG	X	✍️
Sampaio 436	Granbery	36010360	Juiz de Fora	MG	X	✍️

Novo Endereço

Ilustração B. 2 – Tela cadastro de Endereços do SISIMOVEIS

SISTEMA DE AGENDAMENTO IMOBILIARIO

Principal Logar

Cadastramento

Endereço

Cliente

Corretor

Imóvel

Tipo de Imóvel

Status

Cadastramento

Agenda Imóvel

Cadastramento

Agenda Imóvel



Endereço	Tipo de Imóvel	Valor	Cômodos	Metragem	Número	Complemento	Apagar	Alterar
Batista de Oliveira 45	Casa	200.0	4	3	155	apt 402	X	✎
Sampaio 436	Sítio	500.0	8	120	200	bloco1	X	✎

Ilustração B. 3 – Tela cadastro de imóveis do SISIMOVEIS

SISTEMA DE AGENDAMENTO IMOBILIARIO

Principal Logar

Cadastramento

Endereço

Cliente

Corretor

Imóvel

Tipo de Imóvel

Status

Cadastramento

Agenda Imóvel

Cadastramento

Agenda Imóvel



Imóvel	Cliente	Corretor	Data de Agendamento	Hora início	Hora fim	Apagar	Alterar	
Casa	Angelo Cabral	Paulo Cesar	Ana Silva	Janaina Silva	01/05/2013	00:00 hs	00:00 hs	X
Sítio	Angelo Cabral	Paulo Cesar	Ana Silva	Janaina Silva	03/05/2013	00:00 hs	00:00 hs	X

Nova Agenda

Ilustração B. 4 – Tela de cadastro de Agenda de Imóveis do SISIMÓVEIS