

**CENTRO ESTADUAL DE EDUCAÇÃO TECNOLÓGICA
PAULA SOUZA**

**FACULDADE DE TECNOLOGIA DE LINS PROF. ANTÔNIO SEABRA
CURSO SUPERIOR DE TECNOLOGIA EM BANCO DE DADOS**

**CLEBER BORGES DE OLIVEIRA
RICARDO ALBERTO CAMARA HAUY**

**ESTUDO E DESENVOLVIMENTO DE UM SISTEMA DE
INFORMAÇÃO PARA UM ESCRITÓRIO DE ADVOCACIA**

**LINS/SP
2º SEMESTRE 2014**

**CENTRO ESTADUAL DE EDUCAÇÃO TECNOLÓGICA
PAULA SOUZA**

**FACULDADE DE TECNOLOGIA DE LINS PROF. ANTÔNIO SEBRA
CURSO SUPERIOR DE TECNOLOGIA EM BANCO DE DADOS**

**CLEBER BORGES DE OLIVEIRA
RICARDO ALBERTO CAMARA HAUY**

**ESTUDO E DESENVOLVIMENTO DE UM SISTEMA DE UM SISTEMA
DE INFORMAÇÃO PARA UM ESCRITÓRIO DE ADVOCACIA**

Trabalho de Conclusão de Curso apresentado à
Faculdade de Tecnologia de Lins para a obtenção
do Título de Tecnólogo em Banco de Dados.

Orientador: Prof. Mestre Luiz Fernando de Oliveira
Silva

**CLEBER BORGES DE OLIVEIRA
RICARDO ALBERTO CAMARA HAUY**

**ESTUDO E DESENVOLVIMENTO DE UM SISTEMA DE BANCO DE DADOS
PARA UM ESCRITÓRIO DE ADVOCACIA.**

Trabalho de Conclusão de Curso apresentado à
Faculdade de Tecnologia de Lins, como parte dos
requisitos necessários para obtenção do título de
Tecnólogo em Banco de Dados sob orientação do
Prof. Mestre Luiz Fernando de Oliveira Silva.

Data da Aprovação ____/____/____

Orientador: Luiz Fernando de Oliveira Silva

Examinador 1 (nome)

Examinador 2 (nome)

Aos meus pais, Denetis Camargo de Oliveira e Maria Aparecida Borges de Oliveira, a minha irmã Aline Borges de Oliveira e a minha esposa Leticia Rodrigues. Ao meu amigo e parceiro Ricardo e demais amigos pelo apoio e incentivo.

Cleber Borges de Oliveira

**Dedico este trabalho à minha mãe
Izabel Cristina Camara Haury, *in memoriam*, ao
meu pai João Alberto Haury e a todos
estudantes que enfrentam dificuldades.**

Ricardo Alberto Camara Haury

AGRADECIMENTOS

Nessa oportunidade em que é alcançada uma meta importante em nossa vida profissional, queremos expressar os nossos sinceros agradecimentos aos funcionários da instituição FATEC Lins em especial aos professores Me. Luiz Fernando de Oliveira Silva, Me. Adriana de Bortoli, Me. Luciane Noronha do Amaral, Me. Paulo Augusto Nardi, Me. Anderson Pazin, Me. Fábio Lúcio Meira e Me. Alexandre Ponce de Oliveira pelas suas amizades, apoio irrestrito e ajuda tão necessária ao desenvolvimento e realização desse trabalho.

Aos nossos pais e amigos, em especial Lucas Whiather e Laraísa Barbosa que muito nos incentivam e demais professores.

Cleber Borges de Oliveira
Ricardo Alberto Camara Haury

RESUMO

A Tecnologia da Informação (TI) atua em diversas áreas de trabalho, possibilitando uma melhor organização de informações, ajudando a otimizar procedimentos, que antigamente eram feitos de forma manual, e outros, os quais se armazenavam pilhas de papéis, armazenados em um banco de dados, facilitando o acesso remoto às informações, por seus usuários, com ajuda da *Web*. Este estudo tem como objetivo substituir um sistema de um escritório de advocacia que possui algumas limitações, por ser uma aplicação desktop que tem acesso somente local e uma única estação, por um novo software desenvolvido a partir da pesquisa feita com os usuários finais, o que tende a suprir suas necessidades em seu ambiente de trabalho. O aplicativo foi desenvolvido na Plataforma Java, com o armazenamento e gerenciamento ficando por conta do banco de dados Oracle e implementado no *Integrated Development Environment* (IDE) NetBeans. Além disso, foi utilizado *Cascade Style Sheets* (CSS) e *HyperText Markup Language* (HTML) e *Java Server Faces* (JSF) com Primefaces para fazer a visão, Servlets e Beans Gerenciáveis como controladores, e também *Java Database Connectivity* (JDBC) para conexão da aplicação com o banco de dados Oracle, e Hibernate, Astah Community para modelagem na etapa de análise do programa e Data Modeler para modelagem do banco de dados. Trata-se de uma aplicação *Web* que possibilita o acesso remoto às suas atividades por mais de um usuário ao mesmo tempo, o aplicativo armazena dados sobre processos, pessoas envolvidas, honorários, entre outros.

Palavras-Chave: Banco de Dados. *Web*. Java. Oracle. Sistema. Advocacia.

ABSTRACT

The Information Technology (IT) works in different areas of work, enabling better organization of information , helping to optimize procedures that were once made by hand , and others, which were stored piles of papers , stored in a database data communication, enabling remote access to information , by its users, with the Web help. This study aims to replace a system of a law firm that has some limitations because it is a desktop application that has only local access and a single station by a new software developed from the survey of end users , which tends to meet their needs in the workplace . The application was developed in Java Platform , with the storage and management getting because of the Oracle database and implemented the Integrated Development Environment (IDE) NetBeans . In addition , we used Cascade Style Sheets (CSS) and HyperText Markup Language (HTML) and Java Server Faces (JSF) with Primefaces to the vision , Servlets and Beans Manageable as controllers , and also Java Database Connectivity (JDBC) for connection application with the Oracle database, and Hibernate, Astah Community for modeling in program analysis stage and data Modeler for database modeling. This is a web application that allows remote access to its activities for more than one user at the same time , the application stores data about processes , people involved , fees, among others.

Keywords : Database . Web . Java. Oracle. System. Advocacy .

LISTAS DE ILUSTRAÇÕES

Figura 1.1 - Conjunto de Entidades Pessoa	19
Figura 1.2 - Atributo Nome	19
Figura 1.3 - Relacionamento Tem	19
Figura 1.4 - Cardinalidade ou Multiplicidade (mínimo, máximo)	19
Figura 1.5 - Exemplo de um DER	20
Figura 1.6 - Comportamento da Web	28
Figura 1.7 - Representação de uma Classe	33
Figura 1.8 - Relacionamento de Associação Normal	34
Figura 1.9 - Relacionamento de Associação Recursiva	34
Figura 1.10 - Relacionamento de Associação Qualificada	35
Figura 1.11 - Relacionamento de Associação Exclusiva	35
Figura 1.12 - Relacionamento de Associação de Classe Fila	36
Figura 1.13 - Relacionamento de Associação Ternária	36
Figura 1.14 - Relacionamento de Agregação	37
Figura 1.15 - Relacionamento de Generalização/Especialização	37
Figura 1.16 - Diagrama de MVC	38
Figura 1.17 - Diagrama de Sequência	39
Figura 1.18 - Diagrama de Atividade	40
Figura 2.1 - Diagrama de Caso de Uso	43
Figura 2.2 - Diagrama de Entidade e Relacionamento (Modelo Lógico) do	45
Figura 2.3 - Diagrama de Classe do Sistema	46
Figura 2.4 - Diagrama de Atividade Cadastro de Usuário	47
Figura 2.5 - Diagrama de Atividade Listar Todas as Pessoas	47
Figura 2.6 - Diagrama Atividade Alterar Pessoa	48
Figura 2.7 - Diagrama de Atividade Buscar Por Nome	48
Figura 2.8 - Diagrama de Atividade Excluir Pessoa	49
Figura 2.9 - Diagrama de Atividade Exibir Relatório	49
Figura 2.10 - Diagrama de atividades Gerar Despesas	50
Figura 2.11 - Diagrama de Atividades Parcelamento	50
Figura 2.12 - Diagrama de Atividade Andamento de Processos	51
Figura 2.13 - Diagrama de MVC Login do Administrador	51

Figura 2.14 - Diagrama MVC Cadastro Usuário.....	52
Figura 2.15 - Diagrama MVC Pesquisar Usuário	52
Figura 2.16 - Diagrama MVC Alterar Usuário.....	53
Figura 2.17 - Diagrama MVC Excluir Usuário.....	53
Figura 2.18 - Diagrama MVC Gerar Relatório	54
Figura 2.19 - MVC Gerar Despesa.....	54
Figura 2.20 - MVC Parcelamento	55
Figura 2.21 - MVC Andamento do Processo.....	55
Figura 2.22 - Diagrama de Sequência Cadastrar Usuário.....	56
Figura 2.23 - Diagrama de Sequência Alterar Usuário	56
Figura 2.24 - Diagrama de Sequência Excluir Usuário.....	57
Figura 2.25 - Diagrama de Sequência Gerar Relatório	57
Figura 2.26 - Diagrama de Sequência do Andamento de Processo.....	58
Figura 2.27 - Diagrama de Sequência Gerar Despesas.....	58
Figura 2.28 - Diagrama de Sequência gerar e pagar parcelas.....	59
Figura 3.1 - Página Inicial do Sistema.....	60
Figura 3.2 – Tela de Cadastro de Advogados para Acessar o Sistema	60
Figura 3.3 - Tela de Listagem dos Usuários do Sistema.....	61
Figura 3.4 - Tela de Alteração de Usuário pelo Administrador	61
Figura 3.5 - Tela do Módulo Advogado	62
Figura 3.6 – Tela de Cadastro de Pessoas Envolvidas nas Ações	62
Figura 3.7 - Tela Alterando Pessoa.....	63
Figura 3.8 - Tela Tipo de Ações Cadastradas	63
Figura 3.9 - Tela Detalhes da Ação.....	64
Figura 3.10 - Tela de Andamento do Processo	65
Figura 3.11 - Tela de Detalhes da Dívida.....	66
Figura 3.12 - Tela de Inserir Despesas ao Cliente	67
Figura 3.13 - Tela dos Dados de Parcelamento do Cliente.....	67
Figura 3.14 - Tela Relatório de Clientes Inadimplentes.....	68
Figura 3.15 - Tela Relatório de Faturamento Mensal	69

LISTA DE TABELAS

Tabela 2.1 - Necessidades e Prioridades.....	42
--	----

LISTA DE QUADROS

Quadro 1.1 - Modelo Relacional.....	21
-------------------------------------	----

LISTAS DE SIGLAS E ABREVEATURAS

CPF – Cadastro de Pessoa Física
CSS – *Cascading Style Sheets*
DAO - *Data Access Object*
DDL - *Data Definition Language*
DER - Diagrama Entidade Relacionamento
EJB - *Enterprise Java Beans*
GPL - *GNU General Public License*
HTML - *Hypertext Markup Language*
IDE - *Integration Development Environment*
JDBC - *Java Database Connectivity*
JDK7- *Java Development Kit 7*
JEE - *Java Enterprise Edition*
JME - *Java Micro Edition*
JPA- *Java Persistence*
JSE - *Java Standard Edition*
JSF- *Java Server Faces*
JSP - *Java Server Pages*
MER - Modelo Entidade Relacionamento
MIT - *Massachusetts Institute of Technology*
MVC - *Model View Controller*
OO - Orientação a Objetos
OXE10G - *Oracle Express Edition 10g*
PDAs - *Personal Digital Assistant*
PHP - *Hypertext Preprocessor*
PL/SQL - *Procedure Language / Structured Query Language*
RAM - *Random Access Memory*
RG - Registro Geral
SGBD - Sistema de Gerenciamento de Banco de Dados
SQL - *Structured Query Language*
T.I. - Tecnologia da Informação

UML - *Unified Modeling Language*

W3C - *World Wide Web Consortium*

XML - *Extensible Markup Language*

SUMÁRIO

INTRODUÇÃO	14
1 TEORIA E FERRAMENTAS UTILIZADAS	17
1.1 BANCOS DE DADOS	17
1.1.1 Modelagem De Dados	18
1.1.2 MER.....	18
1.2 BANCO DE DADOS ORACLE	22
1.3 LINGUAGENS.....	22
1.3.1 Java	23
1.3.2 HTML	24
1.4 APIs DE PERSISTÊNCIA	25
1.4.1 Hibernate E JPA.....	25
1.4.2 JDBC.....	25
1.5 SCRIPT DE FORMATAÇÃO DE PÁGINAS WEB.....	26
1.5.1 CSS.....	26
1.5.2 JQUERY	26
1.5.3 JAVASCRIPT E AJAX.....	27
1.6 PLATAFORMA DE COMPONENTES WEB.....	27
1.6.1 Servlets e JSP.....	27
1.7 FRAMEWORKS WEB.....	28
1.7.1 JSF E Primefaces	29
1.8 IDEs NETBEANS E IREPORT E FRAMEWORK DE RELATÓRIOS JASPERREPORT	29
1.8.1 NETBEANS.....	29
1.8.2 JASPERREPORTS E IREPORT.....	30
1.9 PADRÃO DE PROJETO	31

1.9.1 Padrão MVC.....	31
1.10 TÉCNICA DE ENGENHARIA DE SOFTWARE.....	32
1.10.1 Análise Orientada a Objeto	32
2 ANÁLISE E PROJETO DO SISTEMA	41
2.1 ANÁLISES DE REQUISITOS.....	41
2.1.1 Análise de Requisitos Funcionais	41
2.2 PERSPECTIVAS DO PRODUTO	42
2.3 CARACTERÍSTICAS	42
2.4 DIAGRAMA DE CASO DE USO	43
2.4.1 Especificação de Histórias	44
2.4.2 Análise de Requisitos Não Funcionais.....	45
2.5 ANÁLISE E DESIGN.....	45
2.5.1 Projeto do Banco de dados.....	45
2.5.1.1 Diagrama de entidade e relacionamento	45
2.6 DIAGRAMA DE CLASSE	46
2.7 DIAGRAMAS DE ATIVIDADES	47
2.8 DIAGRAMAS DE MVC.....	51
3.9 DIAGRAMAS DE SEQUÊNCIA.....	56
3 DISCUSSÕES DOS RESULTADOS	60
CONCLUSÕES	71
REFERÊNCIAS BIBLIOGRÁFICAS	73

INTRODUÇÃO

Os sistemas vêm cada vez mais evoluindo e ocupando parte importante nas organizações. Podemos ver a evolução dos sistemas de banco de dados à medida que as aplicações passariam a utilizar o modelo de dados relacional, fazendo-se o uso de sistemas de banco de dados relacionais. (TEOREY; LIGHTSTONE; NADEAU, 2007)

Pode-se definir sistema como uma junção de partes ou módulos que são interligados e formam um todo se ajudando para cumprir determinado objetivo (OLIVEIRA, 1990). Stair (1998) tem a mesma opinião onde alega que um sistema é um conjunto de elementos ou componentes que interagem para atingir objetivos. Os próprios elementos e as relações entre eles determinam como o sistema trabalha.

Ainda conforme Stair (1998), os sistemas têm entradas, mecanismos de procedimento ou processamento, saídas e *feedback* ou retorno.

Estes sistemas podem melhorar o desempenho das empresas dos dias de hoje quando o assunto a tratar é oportunidade e atuação competitiva, assim sendo facilitado o trabalho dos funcionários e por efeito também os gerentes e administradores passam a compreender a necessidade da utilização desses e os benefícios trazidos à organização. São otimizadas tanto operações internas como externas da empresa, internas no sentido do trabalho em si do dia-a-dia e externas com a facilidade da interligação que se pode fazer desses sistemas com clientes, fornecedores e demais ambientes. (FILHO, 1994)

Com a difusão da área de Tecnologia da Informação (TI) e o surgimento de novos desafios de diversas ciências, o conjunto judiciário vem se adequando a era digital, com o objetivo curto e direto de agregar novos valores e desenvolvimento aos trabalhos judiciários. Com toda essa inovação tecnológica atuando em setores tão tradicionais, o escritório de advocacia não pode ficar alheio a essas mudanças, funcionando com grandes arquivos de metal, repletos de pastas e agendas, para controlar o andamento dos processos, das audiências e o atendimento de seus clientes. (GRANJEIA, 2011)

Assim, com a evolução da tecnologia, hoje em dia até mesmo as pesquisas que os advogados têm que fazer com relação à jurisprudência estão ficando mais fáceis com a utilização de sistemas, pois no Brasil existem duas maneiras de

pesquisar, livros ou sistemas de banco de dados, sendo que em livros limita-se a um raciocínio analítico que tem como desvantagem a fadiga da memória humana, e a demora para encontrar a solução, sendo então importante a área da ciência desenvolver este tipo de tecnologia para satisfazer a necessidade de raciocínio de várias situações passadas rapidamente passando-se a superar essa dificuldade. (LEE, 1998)

O desenvolvimento do sistema do qual trata este trabalho, se dará para um escritório de advocacia, no qual surgem problemas relacionados à defasagem do sistema em vigor tais como o este estar ultrapassado em relação a novas tecnologias, não permitir a emissão de qualquer tipo de relatório necessário e, ainda, o software é incompatível com as versões mais recentes do sistema operacional Windows que é utilizado nos demais computadores do escritório que são máquinas modernas.

Sendo assim, o sistema fica limitado a uma só máquina antiga, não sendo possível trocá-la por uma mais avançada com quatro ou mais gigabytes de *Random Access Memory* (RAM) ou memória de acesso aleatório, pois se trata de uma quantidade de memória incompatível com a versão do Windows XP na qual o sistema está instalado. O acesso ao sistema fica restrito a apenas um usuário, e, por isso, dificulta em muito o desenvolvimento das atividades rotineiras da empresa.

Portanto, este trabalho objetiva cessar as necessidades do escritório no que se refere à implantação e utilização de um sistema de banco de dados confiável e compatível com as tecnologias atuais. Para a realização de tal sistema, pretendeu-se colocar em prática os conhecimentos adquiridos durante o curso de Banco de Dados. Isto se deu pela utilização da linguagem de programação Java para a implementação do padrão de projeto Modelo Visão Controle (MVC) na parte de modelo, e controlador com a tecnologia de *Servlets* para os relatórios ou Beans Gerenciáveis como mediador entre a interface gráfica, o modelo e o acesso ao Banco de Dados, e com o auxílio da linguagem *Hypertext Markup Language* (HTML) ou Linguagem de Marcação de Hipertexto e *Java Server Faces* (JSF) com os componentes adicionais fornecidos pelo *PrimeFaces* que com ajuda da linguagem *Cascading Style Sheets* (CSS) e da biblioteca JavaScript JQuery serão utilizados para a criação da parte de visão. Utilizou-se, para o mapeamento objeto-relacional o *framework Hibernate*, e para consultas a base de dados a linguagem *Hibernate Criteria*.

O trabalho será organizado em três capítulos. No primeiro capítulo será abordado o assunto sobre banco de dados visto que será utilizado banco de dados relacional neste sistema, no qual será discutido o Sistema de Gerenciamento de Banco de Dados (SGBD) escolhido para a aplicação e também tratará das tecnologias referentes às linguagens de programação e ferramentas utilizadas para programar o software. O segundo capítulo tratará da análise e projeto deste novo sistema. Em seguida apresenta-se a conclusão e as referências bibliográficas que foram.

1 TEORIA E FERRAMENTAS UTILIZADAS

1.1 BANCOS DE DADOS

Na década de 60 surgiram os primeiros bancos de dados visto que um computador se tornaria capaz de armazenar permanentemente dados e gerenciá-los com acesso direto e eficiente a estes que, na época, se encontravam em discos magnéticos, e também havia a necessidade de estruturá-los, provendo assim, padrões de acesso aos mesmos. (NASSU; SETZER, 1999)

Segundo Machado (2010, p. 20): “Um banco de dados pode ser definido como um conjunto de dados relacionados. Podemos entender como dados os objetos conhecidos que podem ser armazenados e que possuem um significado implícito”. Daí a criação de Sistemas de Gerenciamento de Banco de Dados (SGBD) que são um conjunto de programas capazes de controlar todas as possíveis necessidades de um banco de dados, desde a declaração de sua estrutura de dados, gravação e leitura destes, recuperação de falhas na comunicação ou na hora que estiver gravando, e controle de concorrência de acesso e de segurança. (NASSU; SETZER, 1999)

Atualmente os sistemas de banco de dados relacionais são os que ocupam maior espaço no comércio, desta forma, afirma-se que:

Embora muitos sistemas de banco de dados especializados (orientados a objetos, espaciais, multimídia, etc.) tenham encontrado substanciais comunidades de usuários nos campos da ciência e engenharia, os sistemas relacionais continuam sendo a tecnologia de banco de dados dominante nas empresas comerciais. (TEOREY; LIGHTSTONE; NADEAU, 2007, p. 2)

Para criar um sistema de banco de dados é necessário seguir alguns passos, estes denominados ciclo de vida e, desta maneira, pode-se entender o que afirma Teorey, Lightstone e Nadeau (2007, p. 3) em: “O ciclo de vida do banco de dados incorpora os passos básicos envolvidos no projeto de um esquema global do banco de dados lógico”.

Desta forma, para se ter um banco de dados de confiança, consistente e de bom desempenho deve-se, em primeiro lugar, passar por uma boa análise de requisitos para que se tenha como saber o modo de desenvolvimento do sistema, neste sentido é mostrada a opinião de Teorey, Lightstone e Nadeau em:

Os requisitos do banco de dados são determinados entrevistando-se os produtores e os usuários dos dados; as informações são usadas para produzir uma especificação formal de requisitos. Essa especificação inclui os dados exigidos para o processamento, os relacionamentos de dados naturais e a plataforma de software para a implementação do banco de dados. (TEOREY; LIGHTSTONE; NADEAU, 2007, p.3).

Em seguida será feita uma modelagem de dados conceitual para descrever como será o sistema.

1.1.1 Modelagem De Dados

A modelagem dos dados é feita utilizando-se o Modelo Entidade-Relacionamento (MER) e criando-se, a partir de suas facilidades, o Diagrama Entidade-Relacionamento (DER) para especificação de requisitos de dados e modelagem conceitual relativos ao banco de dados. (TEOREY; LIGHTSTONE; NADEAU, 2007).

1.1.2 MER

O MER é um modelo semântico que representa a estrutura das informações. Este é apresentado de forma gráfica parecido com um esquema, sendo dessa maneira, uma boa ferramenta de análise de dados. (NASSU; SETZER, 1999)

Logo, o MER oferece um conjunto de formas geométricas e suas denominações para que seja feito a partir dessas formas um diagrama que abstrai a regra que determinado negócio segue para funcionar. Conforme Machado (2010, p.18) afirma: “Abstração, em síntese, nada mais é do que a visão, sem conceituações técnicas, que obtemos na mente de uma realidade qualquer do mundo real”.

Sendo assim, esse modelo é utilizado para criação do Diagrama Entidade-Relacionamento (DER), que vai mostrar ao usuário e às demais pessoas interessadas o significado do sistema, de forma que este modelo é composto basicamente por retângulos que representam conjunto de entidades, círculos que representam atributos, losangos, que por sua vez representam relacionamentos e um par de números entre parênteses que significa a cardinalidade ou multiplicidade

do relacionamento entre as entidades. Isto é o que ilustram as figuras 1.1, 1.2, 1.3 e 1.4 :



Figura 1.1 - Conjunto de Entidades Pessoa
Fonte: Elaborado pelos autores, 2013.



Figura 1.2 - Atributo Nome
Fonte: Elaborado pelos autores, 2013

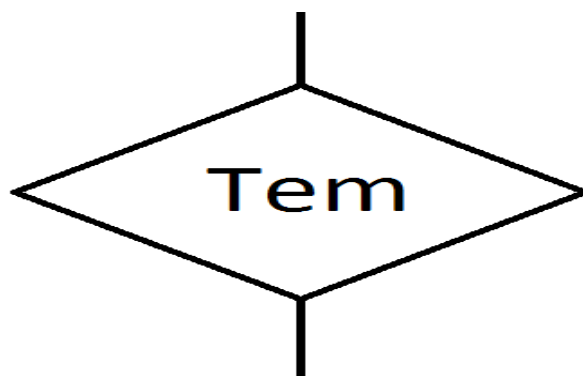


Figura 1.3 - Relacionamento Tem
Fonte: Elaborado pelos autores, 2013.

(1,N)

Figura 1.4 - Cardinalidade ou Multiplicidade (mínimo, máximo)
Fonte: Elaborado pelos autores, 2013.

Com base nessas informações adiante será falado sobre o DER que é o diagrama produzido a partir do MER.

1.1.3 DER

O DER nada mais é do que a representação de um contexto de dados de negócios que devem ser armazenados no banco de dados utilizando-se as formas geométricas oferecidas pelo MER.

Assim afirma Machado (2010, p.16) em: “Modelagem de dados é o estudo das informações existentes em um contexto sob observação para a construção de um modelo de representação e entendimento de tal contexto [...]”.

Deste modo, Machado (2010, p.21) diz que: “É uma descrição de alto nível (macrodefinição), mas que tem a preocupação de captar e retratar toda a realidade de uma organização, processo de negócio, setor, repartição, departamento, etc.”.

A figura 1.5 é um exemplo de um DER.

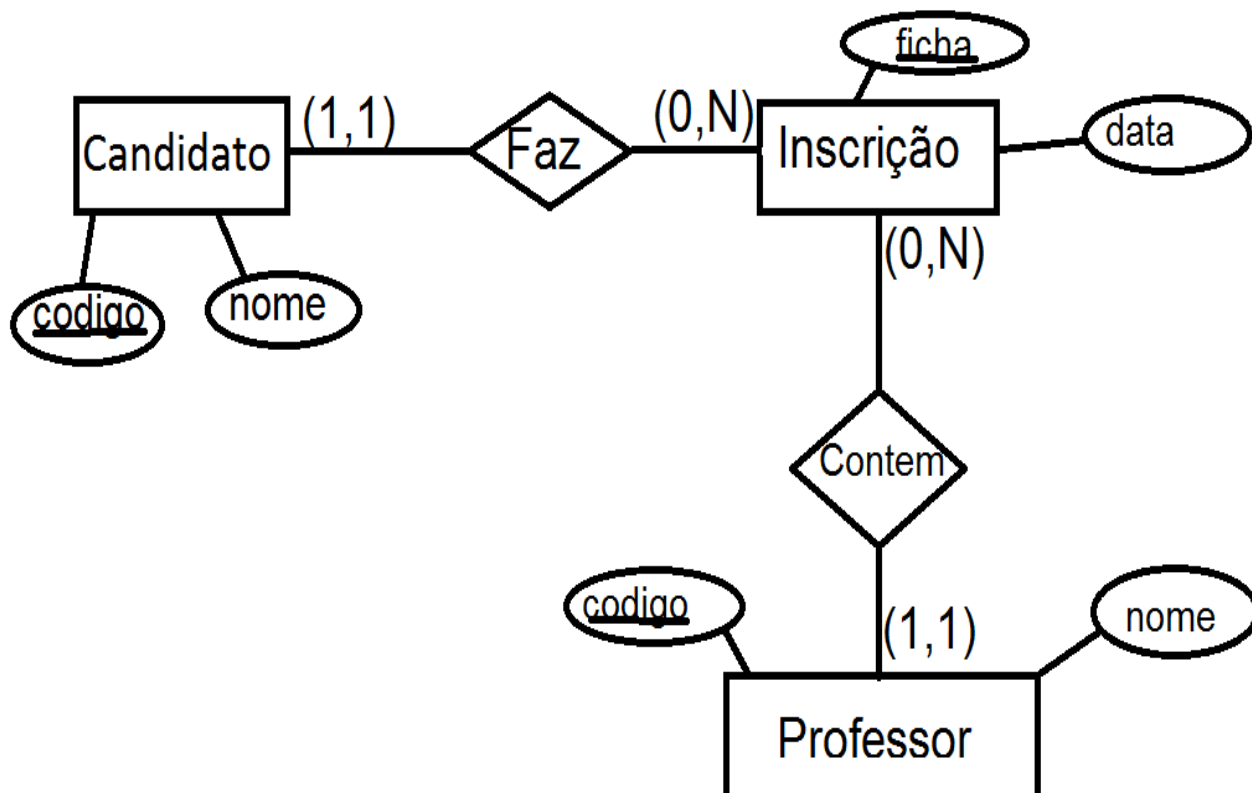


Figura 1.5 - Exemplo de um DER
Fonte: Elaborado pelos autores, 2013.

No DER na figura 1.5 o conjunto de entidades candidato tem como atributos código e nome, sendo o código sua chave primária, ou seja, o atributo capaz de diferenciar uma entidade das demais tornando esta entidade como única.

Percebe-se que o conjunto de entidades Candidato tem um relacionamento binário (relacionamento entre dois conjuntos de entidades) com o conjunto de entidades Inscrição denominado Faz, o qual possui suas cardinalidades ou multiplicidades, o que está na Figura 1.5 significa que cada candidato pode fazer no mínimo zero e no máximo muitas inscrições enquanto que cada inscrição pode ser apenas de no mínimo um e no máximo um candidato.

Em seguida o conjunto de entidades Inscrição possui os atributos número da ficha e data sendo número da ficha sua chave primária.

Por sua vez o conjunto de entidades Inscrição possui relacionamento binário com o conjunto de entidades Professor denominado Contém, ficando assim explícito que, por suas cardinalidades em cada inscrição, só é possível estar um professor tanto no mínimo quanto no máximo e que cada professor pode estar em no mínimo zero e no máximo muitas inscrições. O conjunto de entidades Professor tem dois atributos código (chave primária) e nome.

Para fazer a conversão entre modelos do DER para o modelo relacional é necessário realizar o mapeamento dos dados para este modelo. O modelo relacional é composto de tabelas as quais são criadas com base em oito passos necessários para fazer o mapeamento.

A partir daí tem-se aquilo que talvez, se houver necessidade deverá ser normalizado para ser implementado no banco de dados utilizando Data Definition Language (DDL), ou em português, Linguagem de Definição de Dados, que é a forma que um SGBD dispõe para criar tabelas e relacionamentos, sendo estes, feitos a partir da chave estrangeira no Oracle.

Pode-se ver um modelo relacional relativo ao DER do Quadro 1.1 :

<i>Candidato</i> (<u><i>código</i></u> , <i>nome</i>)
<i>Inscrição</i> (<u><i>ficha</i></u> , <i>códigocandidato</i> , <i>data</i> , <i>códigoprofessor</i>)
<i>Professor</i> (<u><i>código</i></u> , <i>nome</i>)

Quadro 1.1 - Modelo Relacional
 Fonte: Elaborado pelos autores 2013.

O processo de normalização é o último passo antes da implementação do banco de dados no SGBD. Este processo consiste de cinco formas normais, as quais cada uma delas elimina um tipo de dependência de atributos, inconsistência, e demais coisas indesejadas nas tabelas do banco.

Desta maneira quando se tem um esquema de relações já normalizado, basta que sejam passadas para o SGBD utilizando-se os comandos DDL. Agora, será comentado sobre as ferramentas que serão utilizadas para a criação deste sistema.

1.2 BANCO DE DADOS ORACLE

O SGBD escolhido para uso neste sistema foi o *Oracle Express Edition 10g* (OXE10G), esta escolha se justifica por se tratar de um sistema de informação simples para uma pequena empresa, que não ocupará tanto espaço de armazenamento, sendo então desnecessária a compra do software para gerenciamento do banco de dados, lembrando-se que esta versão OXE10G é gratuita e está disponível para *download* no site da Oracle, é um banco de dados com algumas limitações, cujo objetivo principal da empresa Oracle é o ingresso de pequenas empresas a sua Tecnologia. As limitações ficam por conta do tamanho máximo na base de dados que é de 4 gigabytes e ao hardware da máquina utilizada, mais possui os principais recursos do software, e traz a base do Oracle 10g, interface na Web e suporte ao *Procedure Language/Structured Query Language* (PL/SQL) ou linguagem procedural. Para pequenas empresas a versão OXE10G é uma ótima opção já que a versão tem o mesmo suporte a linguagens de programação Oracle 10g integrando-se com Java, C++, PHP entre outros. O Oracle Express possui fórum onde os usuários participam para compartilhar suas experiências e é acompanhada por profissionais do Oracle. (COSTA, 2006)

O Oracle, é uma ferramenta conhecida mundialmente na área da informática no conceito de banco de dados, é um SGBD Relacional e foi um dos primeiros sistemas de banco de dados a utilizar a *Structured Query Language* (SQL) ou, em português, Linguagem de Pesquisa Estruturada. (ABBEY; ABRAMSOM; COREY, 2000)

1.3 LINGUAGENS

1.3.1 Java

Para desenvolver as aplicações utilizou-se a tecnologia Java juntamente com as demais tecnologias a que serão citadas ao longo deste capítulo.

Java é uma linguagem de programação e uma plataforma de computação lançada pela primeira vez pela Sun Microsystems em 1995. É a tecnologia que capacita muitos programas da mais alta qualidade, como utilitários, jogos e aplicativos corporativos, entre muitos outros, por exemplo. O Java é executado em mais de 850 milhões de computadores pessoais e em bilhões de dispositivos em todo o mundo, inclusive telefones celulares e dispositivos de televisão. Há muitos aplicativos e sites que funcionam somente com o Java instalado, e muitos outros aplicativos e sites são desenvolvidos e disponibilizados com o suporte dessa tecnologia todos os dias. O Java é rápido, seguro e confiável. A tecnologia Java está em todo lugar! Ela pode ser encontrada em laptops, datacenters, consoles de jogo, supercomputadores científicos, telefones celulares e até na Internet. (JAVA, 2013, p.1).

Java se justifica por sua facilidade, simplicidade e versatilidade para trabalhar em conjunto com demais formas de trabalho, desenvolvimento e tecnologias.

Segundo Sierra e Bates (2005, p.1): “[...] o Java seduziu os programadores com sua sintaxe amigável, recursos orientados a objetos, gerenciamento de memória e, o melhor de tudo – a promessa de portabilidade”.

O Java chamou a atenção da comunidade de negócios por causa do enorme interesse na Web. O Java é agora utilizado para desenvolver aplicativos corporativos de grande porte, aprimorar a funcionalidade de servidores Web (os computadores que fornecem o conteúdo que vemos em nossos navegadores da Web), fornece aplicativos para dispositivos voltados para o consumo popular (como telefones celulares, pagers e PDAs) e para muitos outros propósitos. (DEITEL; DEITEL, 2010, p. 6).

Os programas Java, por esta linguagem ser orientada a objeto, são compostos de classes, que por sua vez são compostos de atributos e métodos, conhecidos também como características e comportamentos. Existem bibliotecas Java já prontas que são chamadas APIs que estão disponíveis para serem utilizadas por programadores. (DEITEL; DEITEL, 2010).

Esta linguagem possui denominações dentre elas *Java Enterprise Edition* (JEE) e *Java Standard Edition* (JSE) e *Java Micro Edition* (JME).

JSE – *Java Standard Edition* – fornece as funcionalidades essenciais da linguagem e APIs básicas. É nela que está contida as principais classes de uso costumeiro em Java, tais como, String, System e a própria biblioteca

Collection, usada nas nossas aulas. A plataforma JSE oferece muito mais classes para implementar aplicações bem mais sofisticadas, tais como bibliotecas para interfaces gráficas, acesso a arquivos, comunicação em rede, acesso a banco de dados.

JEE – *Java Enterprise Edition* – essa plataforma é usada para o desenvolvimento de aplicações corporativas e cliente-servidor. Uma das principais funcionalidades de JEE atualmente é oferecer tecnologias para o desenvolvimento de aplicações web, tais como: (i) Servlets – que permite definir classes capazes de receber dados de formulários web; (ii) Java Server Pages (JSP) – que possibilita a construção de páginas HTML dinâmicas; e (iii) Enterprise Java Beans (EJB) – que permite definir classes que oferecem serviços distribuídos que contêm as regras de negócio principal da aplicação.

JME – *Java Micro Edition* – É a plataforma de classes Java voltada para os dispositivos com capacidade reduzida de processamento e armazenamento de dados, tais como aparelhos celulares, palms, etc. Ela é dessa forma usada para desenvolvimento das aplicações denominadas de “embarcada” que está presente em brinquedos, robôs, eletrodomésticos, etc. (BEZERRA; GOMES, 2013, p.1)

1.3.2 HTML

Há mais de 20 anos atrás foi criada a linguagem de HTML, ainda no início da nova linguagem, ela foi submetida a ligeiras modificações de texto simples, introduzindo imagens e adicionando a ferramenta CSS para formatar textos.

Após alguns anos de criação da linguagem HTML, foi lançada uma nova versão HTML4, seria essa uma nova tecnologia um grande avanço na linguagem HTML, mas isso foi em 1997 quando ainda tínhamos limitações em computadores e seus processos, e também a conexão de internet que era discada e muito lenta.

Com a evolução da internet que passou a ser comercializada e medida em megabytes, houve a necessidade de procurar por novas tecnologias de linguagem, foi então que surgiu o novo e ambicioso projeto de um novo padrão HTML.

O HTML5, que foi planejado mais não era visto com bons olhos por muitos especialistas da área da informática, um projeto ambicioso que foi desenvolvido para substituir o antigo padrão HTML com a finalidade agilizar tanto a sua área de programação quanto a utilização do usuário, sendo uma versão com mais componentes e novas ferramentas próprias para a utilização de imagens, áudio e vídeo. Mesmo sendo um projeto não confiável por muitos o projeto vingou e foi adotado primeiramente pelo navegador Firefox, e rapidamente seguido por vários navegadores.

A linguagem HTML5 tem uma vasta tecnologia gráfica o que facilita a sua necessidade de utilização e proporciona mais liberdade de desenvolvimento gráfico, sendo assim podendo fazer a diferença no produto final. (DAVID, 2010)

1.4 APIs DE PERSISTÊNCIA

1.4.1 Hibernate E JPA

O Hibernate é um conjunto de ferramentas para o mapeamento objeto-relacional desenvolvido em Java, que consistem e transformar objetos definidos pelo desenvolvedor em dados tabulares de uma base de dados, como o Hibernate o programador economiza tempo, pois deixa de ter que se preocupar com a programação do banco de dados, e ainda se for preciso trocar de banco de dados não é necessário ele ficar alterando seu código, apenas necessita informar o novo banco. O Hibernate coleta informações das classes Java e as transforma em tabelas gerando o código SQL de acordo com o Banco de Dados que será utilizado. (ORTONCELLI; LAZARIN, 2009)

“Após o sucesso do Hibernate, a especificação JPA (*Java Persistence API*) foi criada com o objetivo de padronizar as ferramentas ORM para aplicações Java e consequentemente diminuir a complexidade do desenvolvimento.” (K19, 2014, p.45)

O JPA é a mais recente especificação criada para permitir persistência objeto-relacional com bancos de dados relacionais. Ao invés de salvar dados em tabelas, o código do sistema pede pela persistência de classes carregadas com os valores que se quer salvar. Assim, os produtos que implementam JPA, como Hibernate, transformam as requisições de salvar ou consultar via classes, em comandos SQL que são enviados ao banco de dados. (GOMES, 2008, p.11)

1.4.2 JDBC

Para se pensar em criar um sistema de banco de dados deve-se pensar em como conectar-se a este banco de dados.

A *Java Database Connectivity* (JDBC) é uma intermediária entre o banco de dados e a aplicação Java responsável por enviar instruções e receber resultados a partir de um conjunto de classes Java criadas para esse propósito.

SGBD diferentes possuem dialetos ou estrutura diferente uns dos outros, a JDBC provê a comunicação da aplicação com o SGBD independentemente de estrutura ou dialeto por estar implementadas classes para suportar mais de um tipo ou os principais tipos de SGBD. (BRILL, 2001)

Neste sistema será utilizado o *Oracle Driver* que é o driver para conexão e uso do SGBD Oracle.

1.5 SCRIPT DE FORMATAÇÃO DE PÁGINAS WEB

1.5.1 CSS

As CSS são uma boa maneira de estilizar, e modelar uma página HTML. Em setembro de 1994, surgiu a primeira proposta para implementação das CSS, e no mês de dezembro de 1996, as CSS1 foram lançadas como uma Recomendação oficial do W3C. (SILVA, 2012)

CSS veio para devolver a HTML sua função principal que é a marcação de texto, sendo assim CSS vem com a função de apresentação de conteúdo, utilizando-se de suas funções para a formatação de textos e posicionamentos ao conteúdo HTML. A CSS também facilitou na exibição do conteúdo por ser compatível com diversos navegadores da Web. (SILVA, 2012, p. 24)

Ou seja, as CSS põe um fim na estilização *in-line* em tags HTML em linha de comando conforme se fazia anteriormente ou dentro do documento ou arquivo CSS.

1.5.2 JQUERY

A JQuery é uma biblioteca escrita em JavaScript que foi criada para simplificar esta linguagem para que se conseguisse obter os mesmos resultados com menos linhas de códigos.

JQuery é uma biblioteca JavaScript criada por John Resig e disponibilizada como software livre e aberto, ou seja, de emprego e uso regido segundo licença conforme as regras estabelecidas pelo MIT (Massachusetts Institute of Technology) e pelo GPL (GNU General Public License). Isto, resumidamente significa que você pode usar a biblioteca gratuitamente tanto para desenvolver projetos pessoais como comerciais. (SILVA, 2006, p.20).

JQuery é composta de métodos que substituem aqueles comandos complicados da linguagem JavaScript.

Simplicidade é a palavra-chave que resume e norteia o desenvolvimento com JQuery. Linhas e mais linhas de programação JavaScript escritas para obter um simples efeito em um objeto são substituídas por apenas algumas, escritas com sintaxe JQuery. Intrincados e às vezes confusos códigos JavaScript destinados a selecionar um determinado elemento HTML componente da árvore do documento são substituídos por um simples método JQuery. (SILVA, 2006, p.26)

Não é necessário um conhecimento de JavaScript para desenvolver com JQuery, basta aprender a utilizar seus métodos. (SILVA, 2006).

1.5.3 JAVASCRIPT E AJAX

JavaScript foi criada pela Netscape em parceria com a Sun Microsystems, com a finalidade de fornecer um meio de adicionar interatividade a uma página web. A primeira versão, denominada JavaScript 1.0, foi lançada em 1995 e implementada em março de 1996 no navegador Netscape Navigator 2.0 quando o mercado era dominado pela Netscape. (SILVA, 2010, p.23)

Foi escolhido utilizar o conjunto de ferramentas Ajax com o intuito de melhorar interatividade com o usuário, através de sua solicitação assíncrona de informações, possibilitando fazer uma nova solicitação ao servidor web sem que tenha que recarregar a mesma página.

A palavra Ajax vem da expressão Asynchronous JavaScript and XML. É o uso sistemático de JavaScript e XML (entre outras tecnologias) para tornar o navegador mais interativo com o usuário, utilizando-se solicitações assíncronas de informações. (NIEDERAUER, 2013, p.15).

1.6 PLATAFORMA DE COMPONENTES WEB

1.6.1 Servlets e JSP

Os Servlets são da plataforma *Java Enterprise Edition* (JEE) e funcionam como requisições para o servidor que devolve uma resposta, esta resposta pode ser um documento em *Java Server Pages* (JSP), assim se torna possível a criação e utilização de conteúdo dinâmico para as estáticas páginas Web em HTML.

Pode-se dizer que o Servlet controla o que deve ser requisitado enquanto que o JSP (que nada mais é que Java dentro junto com comandos HTML) traz aquilo que foi pedido para a visão do usuário.

A aplicação que roda no servidor disponibiliza somente páginas estáticas, porém, uma outra aplicação “assistente”, com a qual o servidor pode se comunicar, pode construir páginas dinâmicas instantaneamente. Uma página dinâmica poderia ser qualquer coisa, desde um catálogo até um weblog, ou até mesmo uma página que escolha imagens aleatoriamente para exibi-las. (BASHAM; SIERRA; BATES, 2008, p. 26).

Na Figura 2.1 têm-se os clientes que nada mais são que web *browsers* ou navegadores localizados em várias localidades diferentes e servidores também localizados em lugares distintos, os clientes é que são responsáveis por fazerem requisições (chamadas de aplicações dinâmicas, ou seja, Servlets) e os servidores de receber e depois respondê-las (JSP – Java em páginas com código HTML).

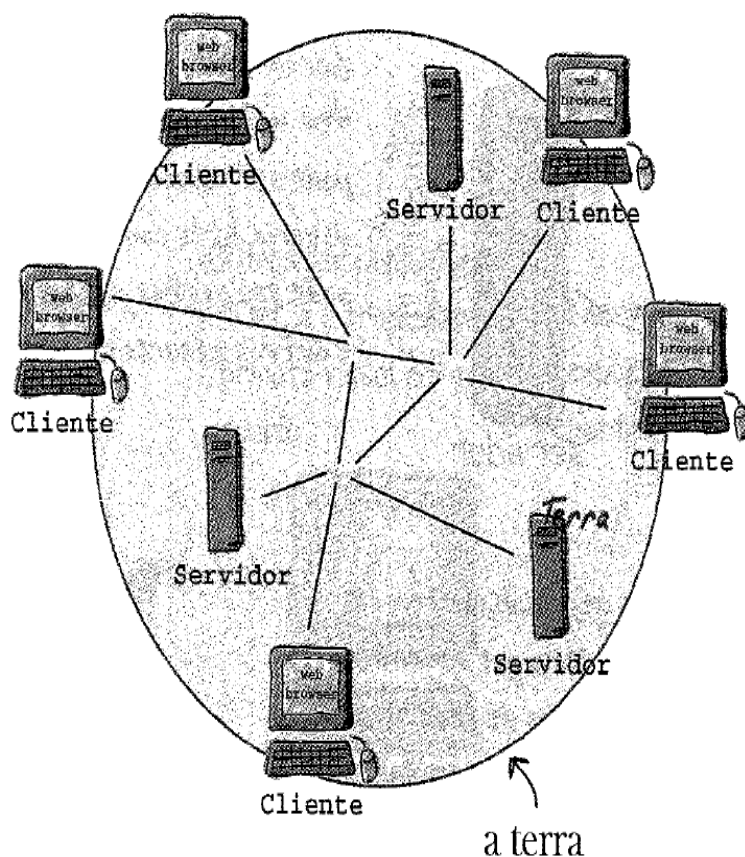


Figura 1.6 - Comportamento da Web

Fonte: Basham; Sierra; Bates, 2008, p.3.

1.7 FRAMEWORKS WEB

1.7.1 JSF E Primefaces

A ferramenta JSF nos permite criar aplicações Java para Web utilizando componentes visuais praticamente pronto, com isso faz com que o desenvolvedor não se preocupe com outros tipos de códigos e tecnologias para implementar formulários, tabelas calendários entre outros objetos.

O JSF (Java Server Faces) é tecnologia padrão do J2EE 1.4 (ou superior) para criar aplicações web. Ele herda das tecnologias JSP e Servlets e estende seus conceitos com um ciclo de vida e um conjunto de componentes e recursos sofisticados e focados no desenvolvimento RAD para web. (GOMES, 2008, p.11)

Projetado para ser flexível, a tecnologia JavaServer Faces utilizada, UI e web-tier conceitos padrão existente sem limitar aos desenvolvedores uma linguagem particular de mark-up, protocolo ou dispositivo cliente.[..] A facilidade de utilização é o objetivo principal, o JavaServer Faces arquitetura define claramente uma separação entre a lógica da aplicação e apresentação enquanto tornando mais fácil para ligar a camada de apresentação para o código da aplicação. Este projeto permite que cada membro de uma equipe de desenvolvimento de aplicações web para se concentrar em sua parte do processo de desenvolvimento, e também fornece um modelo de programação simples para conectar as peças.[..] (JUNEAU, 2014, p.1).

Com a utilização da biblioteca PrimeFaces será dado enriquecimento aos conteúdos gráficos do sistema, sendo que ele possui diversos objetos como calendários botões e tabelas, que já estão prontos e precisamos apenas invocar mediante chamada de tags, com isso temos a simplificação da interface gráfica.

PrimeFaces é uma biblioteca de componentes de código aberto para o JSF, que possui diversos componentes, vem se destacando diante de outras bibliotecas do mesmo gênero, devido sua boa documentação. (JUNEAU, 2014)

1.8 IDEs NETBEANS E IREPORT E FRAMEWORK DE RELATÓRIOS JASPERREPORT

1.8.1 NETBEANS

A *Integration Development Environment* (IDE) NetBeans possibilita a implementação da aplicação JAVA, HTML, CSS, JQUERY, e JDBC para comunicação com o Banco de Dados e o servidor com a base na tecnologia SERVLET e JSP, entre outros tipos de tecnologias. Com a utilização da IDE NetBeans no decorrer das aulas concluímos que é um aplicativo que agrega ferramentas e recursos que auxiliam muito na criação de programas.

O NetBeans é um ambiente de desenvolvimento, *open source* escrito totalmente em Java. A IDE NetBeans vem evoluindo rapidamente a cada versão, tornando-se competitiva junto as melhores ferramentas de desenvolvimento Java do mundo. (GONÇALVES, 2006, p.23)

Com o NetBeans é possível o desenvolver aplicações em um curto tempo pois tem diversos atalhos e ferramentas que auxilia o desenvolvedor em sua IDE, possibilita o desenvolvimento de aplicações desktop Java, moveis e Web, oferecendo excelentes ferramentas de PHP e C/C++, além de ser um IDE de código aberto e gratuito, é o pioneiro em suporte a JDK7, Java EE6 e JavaFX. O NetBeans é um editor Java que se encontra em constante desenvolvimento, e este suporta várias linguagens como Java, C/C++, XML, HTML, PHP, Groovy, Javadoc, JavaScript e JSP e possui compatibilidade em todos os sistemas operacionais que suportem Java. (GONÇALVES, 2006).

1.8.2 JASPERREPORTS E IREPORT

“JasperReports é um framework para geração de relatórios. É uma ferramenta totalmente open source e gratuita, e a mais utilizada com esse proposito atualmente[...]”.(MACEDO, 2010).

Uma característica importante do JasperReports é que seu layout de relatório é um arquivo XML, que possui a extensão .xml, com isso o XML contem todo conteúdo de formatação do relatório e também possui os campos que serão preenchidos no decorrer de seu uso. Com o usos do JasperReports podemos exportar relatórios para diversos formatos, tais como PDF, HTML, XML,XLS entre outros. (MACEDO, 2010, p.1)

O *iReport* é uma ferramenta desenvolvida pela mesma empresa do *JasperReports*, a *JasperForge*, e por isso é muito comum ver os dois

sendo usados em conjunto. Uma das dificuldades ao trabalhar com os relatórios, está na definição do *layout*. É complicado escrever o layout totalmente em XML, sem ter que se aprofundar em todas as *tags* e atributos possíveis, e além disso posicionar todos os elementos corretamente. Na prática, é muito raro alguém editar o JRXML manualmente, e sim apenas para fazer alguns pequenos ajustes quando necessários. O processo normal é utilizar alguma ferramenta para gerar o JRXML automaticamente, e o *iReport* é utilizado com esse propósito. (MACEDO, 2010, p.1)

Com a utilização do iReport é possível o desenvolvedor economizar tempo, já que o mesmo é um aplicativo gráfico que permite a modelagem do relatório utilizando uma paleta e arrastando e soltando seus componentes, o iReport traz consigo um conjunto de *templates* que pode ser utilizado diretamente. (MACEDO, 2010)

1.9 PADRÃO DE PROJETO

1.9.1 Padrão MVC

O padrão de programação ou organização a ser utilizado tem o nome de *Model View Controller* (MVC) ou em português Modelo Visão e Controlador. Este padrão separa as classes em pacotes diferentes cada um relacionado ao que a classe faz parte se é do modelo, visão ou do controlador. Em casos atuais cria-se também um pacote chamado *Data Access Object* (DAO) ou Objeto de Acesso a Dados, banco de dados para ser mais exato, neste pacote estarão todas as classes Java de acesso, conexão e demais necessidades tais como busca, alteração, inserção, exclusão, etc.

O modelo (Model) é utilizado para manipular informações de forma mais detalhada, sendo recomendado que, sempre que possível, se utilize dos modelos para realizar consultas, cálculos e todas as regras de negócio do nosso site ou sistema. É o modelo que tem acesso a toda e qualquer informação sendo essa vinda de um banco de dados, arquivo XML. Visão (view) é responsável por tudo que o usuário final visualiza, toda a interface, informação, não importando sua fonte de origem, é exibida graças a camada de visão.

A Controladora (controller), como o nome já sugere, é responsável por controlar todo o fluxo de informação que passa pelo site/sistema. É na controladora que se decide “se”, “o que”, “quando” e “onde” deve funcionar. Define quais informações devem ser geradas, quais regras devem ser

acionadas e para onde as informações devem ir, é na controladora que essas operações devem ser executadas. Em resumo, é a controladora que executa uma regra de negócio (modelo) e repassa a informação para a visualização. (BASTOS, 2011, p. 1)

1.10 TÉCNICA DE ENGENHARIA DE SOFTWARE

1.10.1 Análise Orientada a Objeto

A Orientação a Objetos (OO) nada mais é que um paradigma (modelo) de programação no qual se tem o conceito de classes como objetos do mundo real ou abstratos, estes possuindo atributos ou características e métodos ou comportamentos. Na OO utiliza-se os conceitos de herança e polimorfismo.

Segundo RICARTE (2001) a uma característica comum do mecanismo Herança é que a mesma permite que diversas subclasses sejam criadas a partir de uma classe base ou superclasse. Existem algumas formas de relacionamento de Herança, como a extensão que é uma subclasse que acrescenta membros a uma superclasse como atributos e métodos, já a especificação como o próprio nome já diz apenas especifica o que uma subclasse deve oferecer, e também existe a combinação das Heranças de Extensão e Especificação tem a finalidade de uma classe com característica de extensão que é uma subclasse ter uma especificação. Lembrando que a classificação de Especificação em Herança não pode realizar nenhuma alteração seja em classe base, superclasse ou mesmo subclasse.

Já com relação a polimorfismo entende-se como muitas formas ou mais de uma forma.

Polimorfismo é o princípio pelo qual duas ou mais classes derivadas de uma mesma superclasse podem invocar métodos que têm a mesma identificação (assinatura) mas comportamentos distintos, especializados para cada classe derivada, usando para tanto uma referência a um objeto do tipo da superclasse. Esse mecanismo é fundamental na programação orientada a objetos, permitindo definir funcionalidades que operem genericamente com objetos, abstraindo-se de seus detalhes particulares quando esses não forem necessários. (RICARTE, 2001, p. 6)

A análise orientada a objeto nada mais é que uma forma de especificação das classes, atributos e métodos (operações) de uma forma por diagramas que constituem uma linguagem chamada *Unified Modeling Language* (UML) na qual

descreve-se todo o comportamento de um programa desenvolvido a partir da orientação a objetos.

Os componentes da análise orientada a objetos são diagramas de classes, diagrama de sequência, diagrama de MVC, diagrama de atividades, e diagrama de estados.

No diagrama de classes, as classes são representadas por retângulos, os atributos por nome e tipo de atributo, os métodos pelo nome e assinatura e o relacionamento entre as classes por associações entre elas. As associações podem ser normais, recursivas, qualificada, exclusiva, ordenada (coloca-se apenas a palavra ordenada na linha de associação), de classe, ternária, agregação, generalização ou especialização.

Na figura 1.7 é ilustrada uma classe e suas descrições que são o nome, atributos e métodos (Operações).

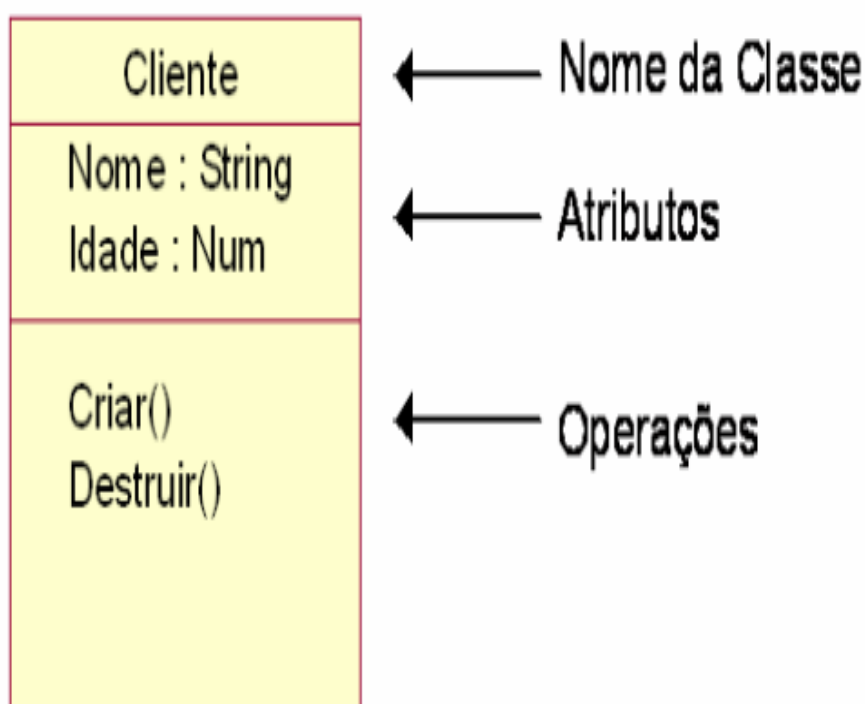


Figura 1.7 - Representação de uma Classe
Fonte: Silva, 2013, p5.

Na figura 1.8 é ilustrado o Relacionamento entre duas classes que possuem um relacionamento de associação normal, indicando que o cliente possui conta corrente e que a conta corrente pertence ao cliente.

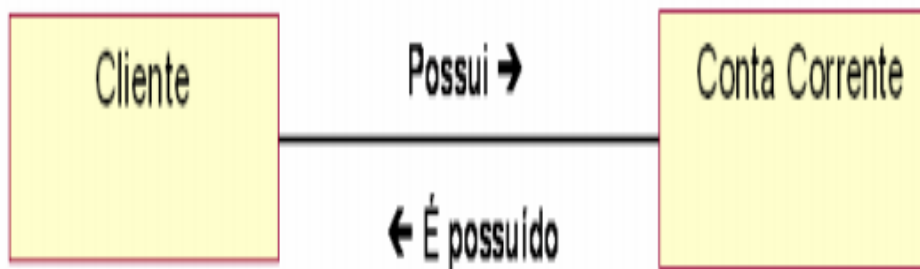


Figura 1.8 - Relacionamento de Associação Normal

Fonte: Silva, 2013, p8.

Na figura 1.9 é ilustrado o Relacionamento de Associação Recursiva que é quando a classe necessita de uma conexão com ela mesma representando de forma diferente a ligação entre os objetos no caso pessoa pode ser um marido que é casado uma esposa, sendo que esposa também é herdada da classe pessoa.



Figura 9 1.9 - Relacionamento de Associação Recursiva

Fonte: Silva, 2013, p8.

Na figura 1.10 é ilustrado um Relacionamento de Associação Qualificada, utilizada para indicar um para muitos (1..*) ou de muitos(*), "O identificador é desenhado com uma pequena caixa fina de associação". (Silva.2013.p8))

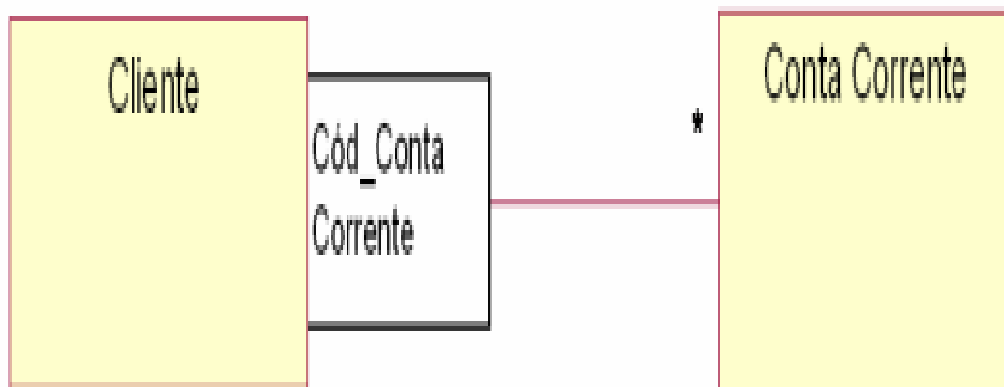


Figura 1.10 - Relacionamento de Associação Qualificada
 Fonte: Silva, 2013, p9.

Na figura 1.11 é ilustrado o Relacionamento de Associação Exclusiva, onde uma classe é exclusiva de outra classe, ou seja, uma pessoa possui um contrato, ou a empresa possui um contrato, nessa representação não é permitida e não existe uma relação entre pessoa e empresa para possuírem um único contrato.

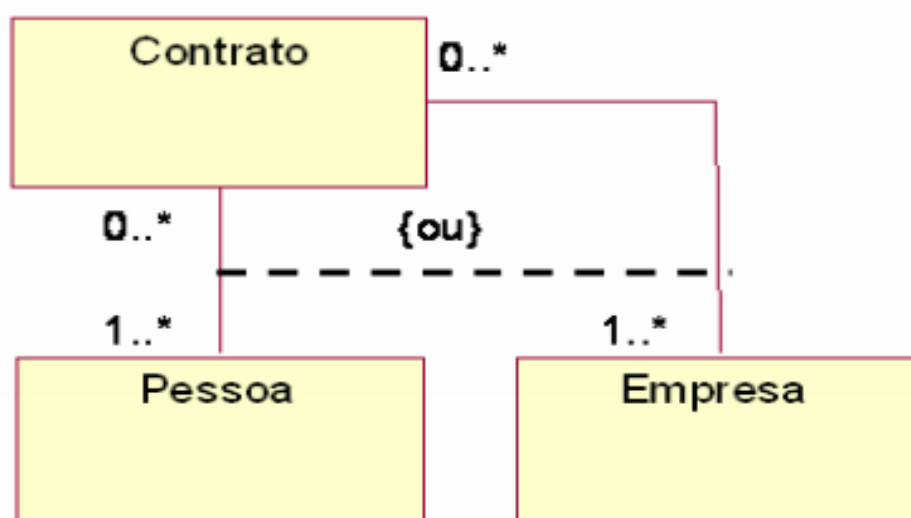


Figura 1.11 - Relacionamento de Associação Exclusiva
 Fonte: Silva, 2013, p10.

Na figura 1.12 é ilustrado o Relacionamento de Associação de Classe Fila, onde uma terceira classe é ligada com linha tracejada a linha de relacionamento de outras duas classes, facilitando a identificação da classe fila e podemos observar que fica fácil a sua identificação sendo que a classe fila possui a linha tracejada.

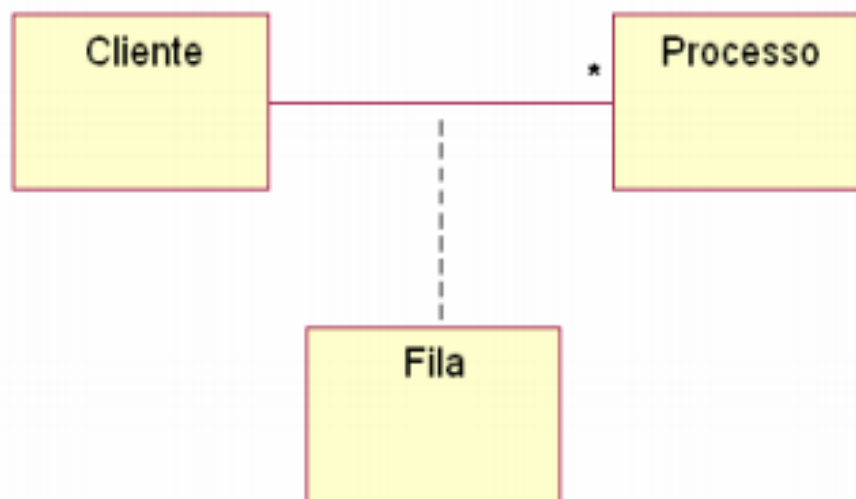


Figura 1.12 - Relacionamento de Associação de Classe Fila
Fonte: Silva, 2013, p11.

Na figura 1.13 é ilustrado o Relacionamento de Associação Ternária que possui três classes associadas entre si, na representação mostra a relação em que, o cliente pode possuir um ou mais contratos e que cada contrato possui uma ou mais regras contratuais.

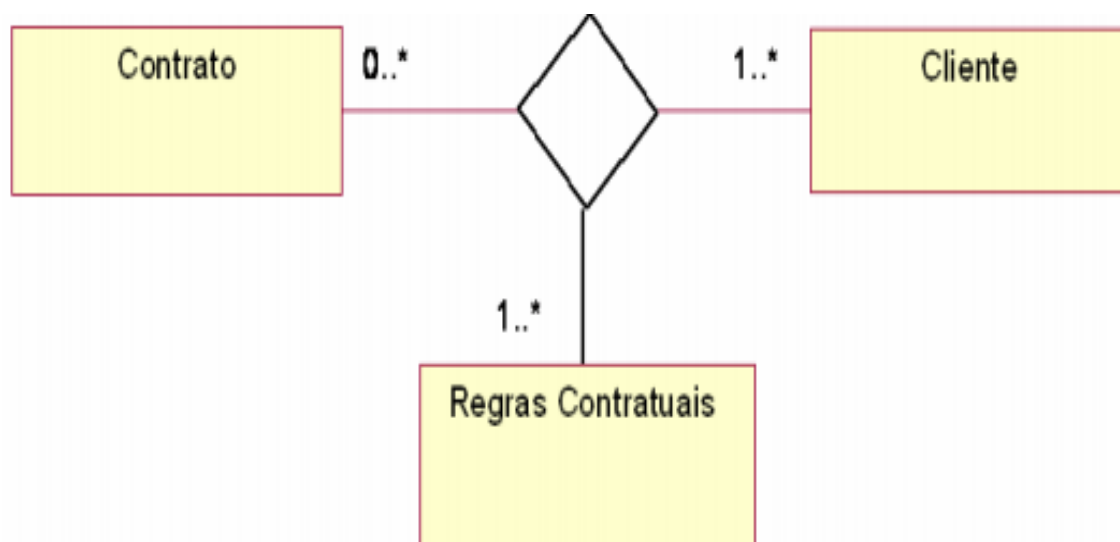


Figura 1.13 - Relacionamento de Associação Ternária
Fonte: Silva, 2013, p12.

Na figura 1.14 é ilustrado o Relacionamento de Associação de Agregação onde indica que uma classe no relacionamento é contida ou é parte de outra classe, é possível identificar a agregação de acordo com as palavras chaves: “consiste em”, “contém” e “é parte de”, na ilustração entende-se que a Marinha contém Navios.

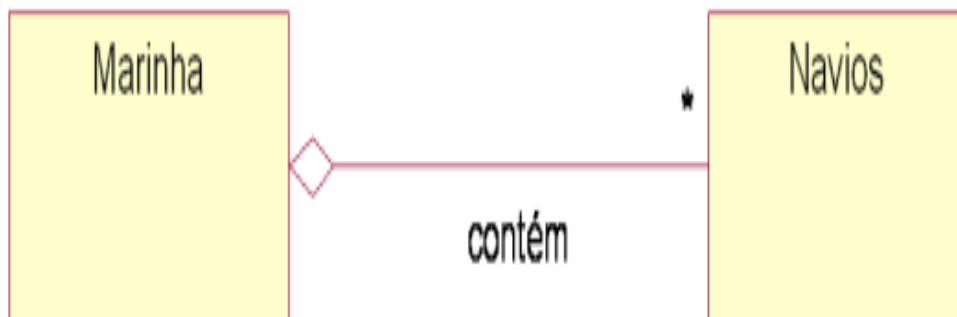


Figura 1.14 - Relacionamento de Agregação
Fonte: Silva, 2013, p13.

Na figura 1.15 é ilustrado o Relacionamento de Generalização ou Especialização é onde representa o relacionamento entre uma classe geral e outras específicas como na ilustração onde pessoa é a classe geral e homem e mulher são classes específicas mais não deixam de ser uma classe pessoa o q diferencia é que na classes específicas contém atributos mais detalhados.

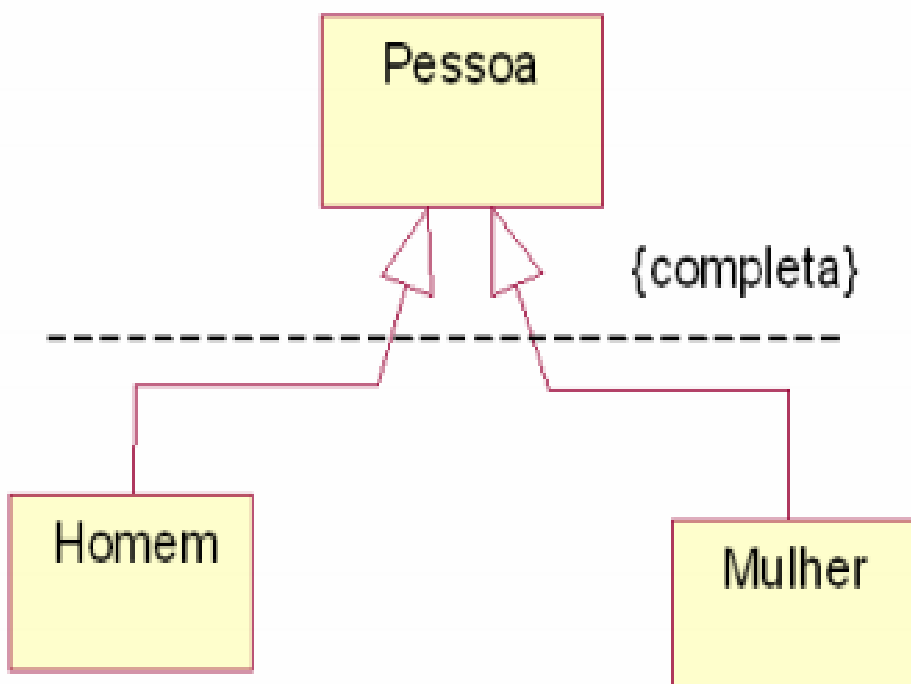


Figura 1.15 - Relacionamento de Generalização/Especialização
Fonte: Silva, 2013, p17.

No diagrama de MVC são separadas as classes do modelo, visão e controlador, e também a classe que representa o acesso ao banco de dados,

representa-se também os objetos de uma classe instanciados em outras, a partir deste diagrama se constrói o diagrama de sequência.

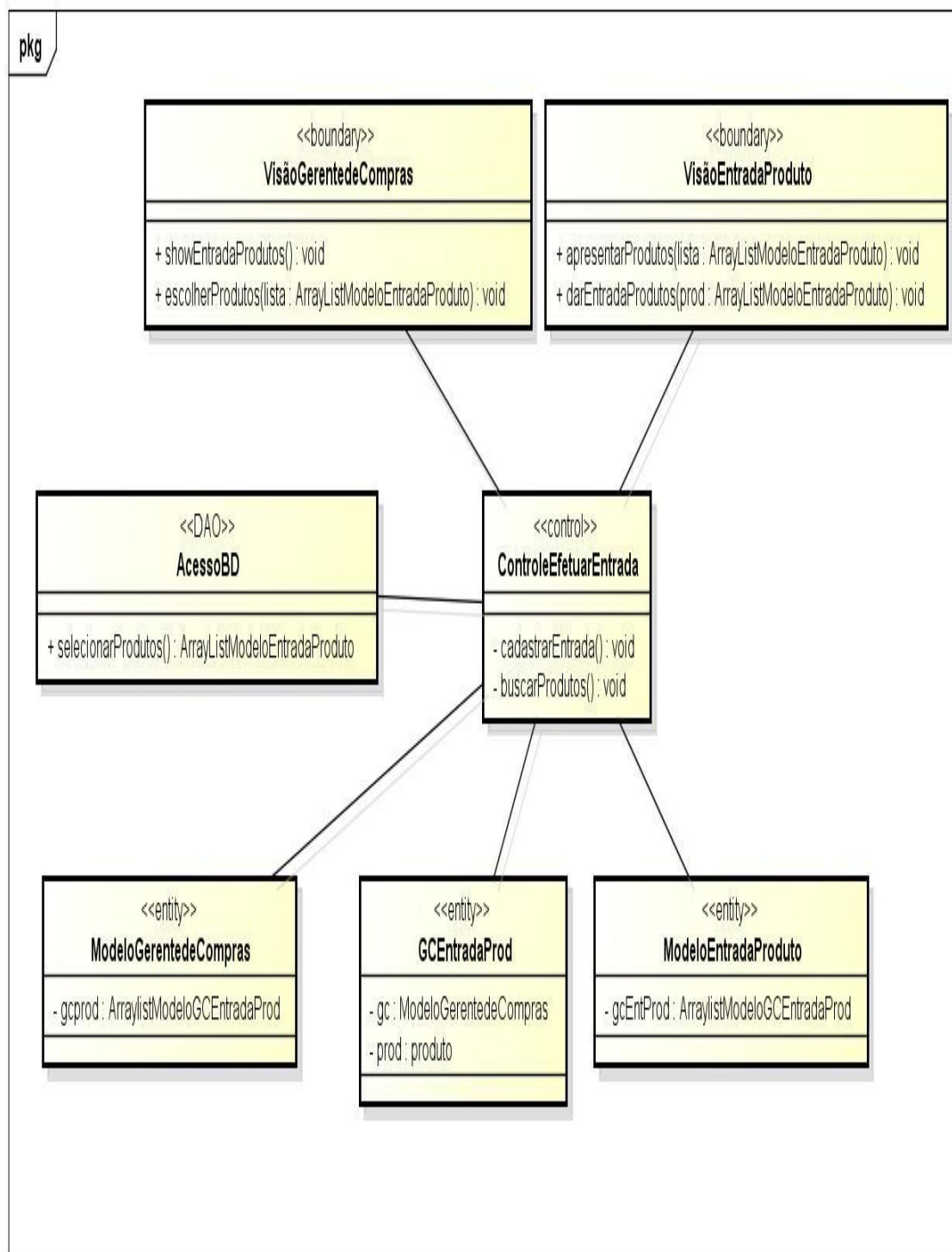


Figura 1.16 - Diagrama de MVC
Fonte: Elaborado pelos autores, 2013.

De outra forma, no diagrama de sequência é descrita a sequência em que acontecem as chamadas dos métodos e a que classe estão associados, neste diagrama representa-se as classes com círculos e os métodos chamados com setas em direção a classe deste de acordo com a linha do tempo que é indicada de cima para baixo.

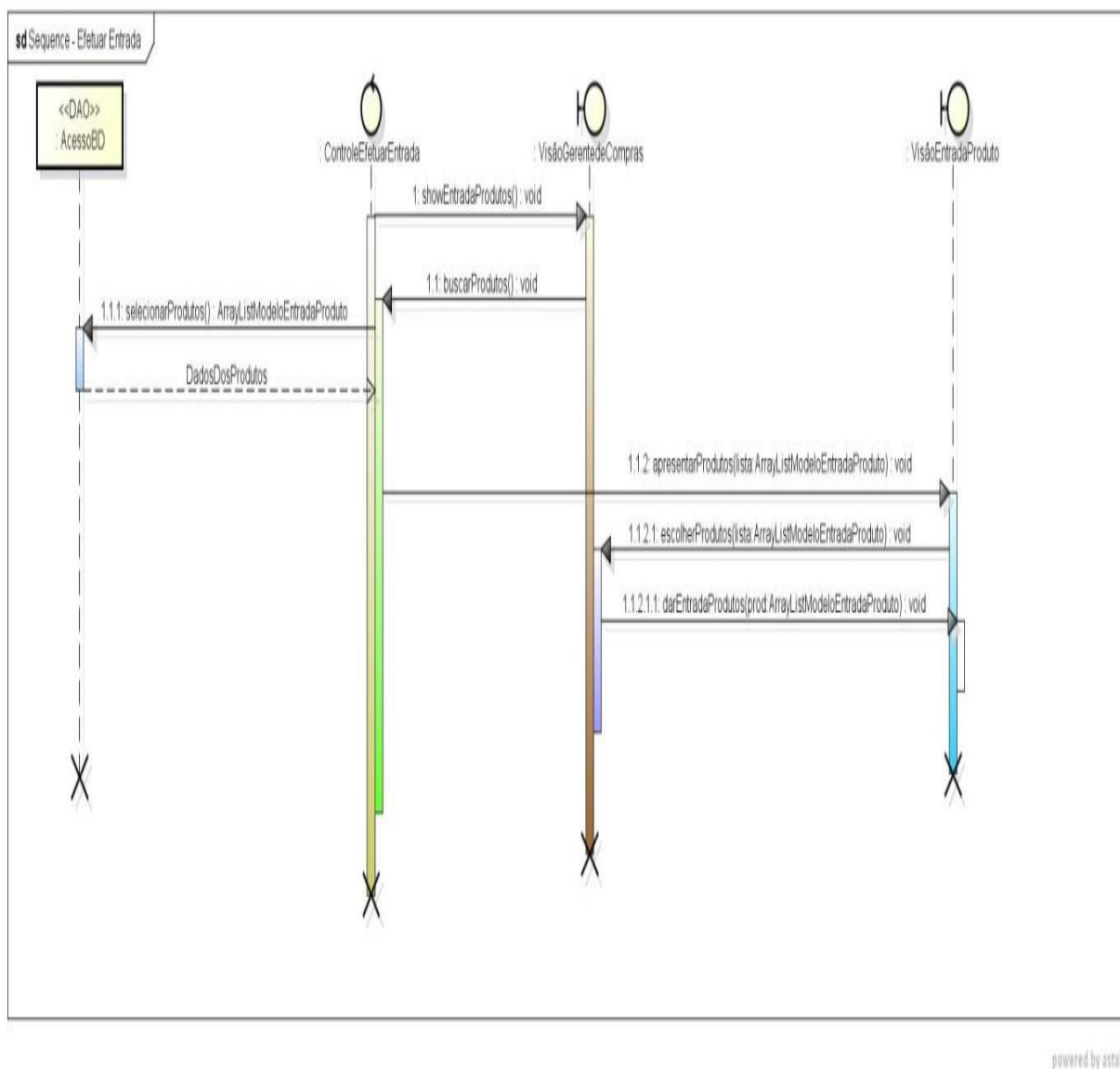


Figura 1.17 - Diagrama de Sequência

Fonte: Elaborado pelos autores, 2013.

Por fim, no diagrama de atividades são descritas todas as atividades que podem ser feitas pelo sistema, bem como todos os caminhos que se pode tomar e até onde eles vão terminar. Aqui, as atividades são representadas em forma de um retângulo com os cantos ovais, os repositórios de dados por retângulos e a

sequência das atividades por setas de uma atividade para outra, os triângulos são junções de mais de uma atividade ou decisões para qual atividade deve seguir.

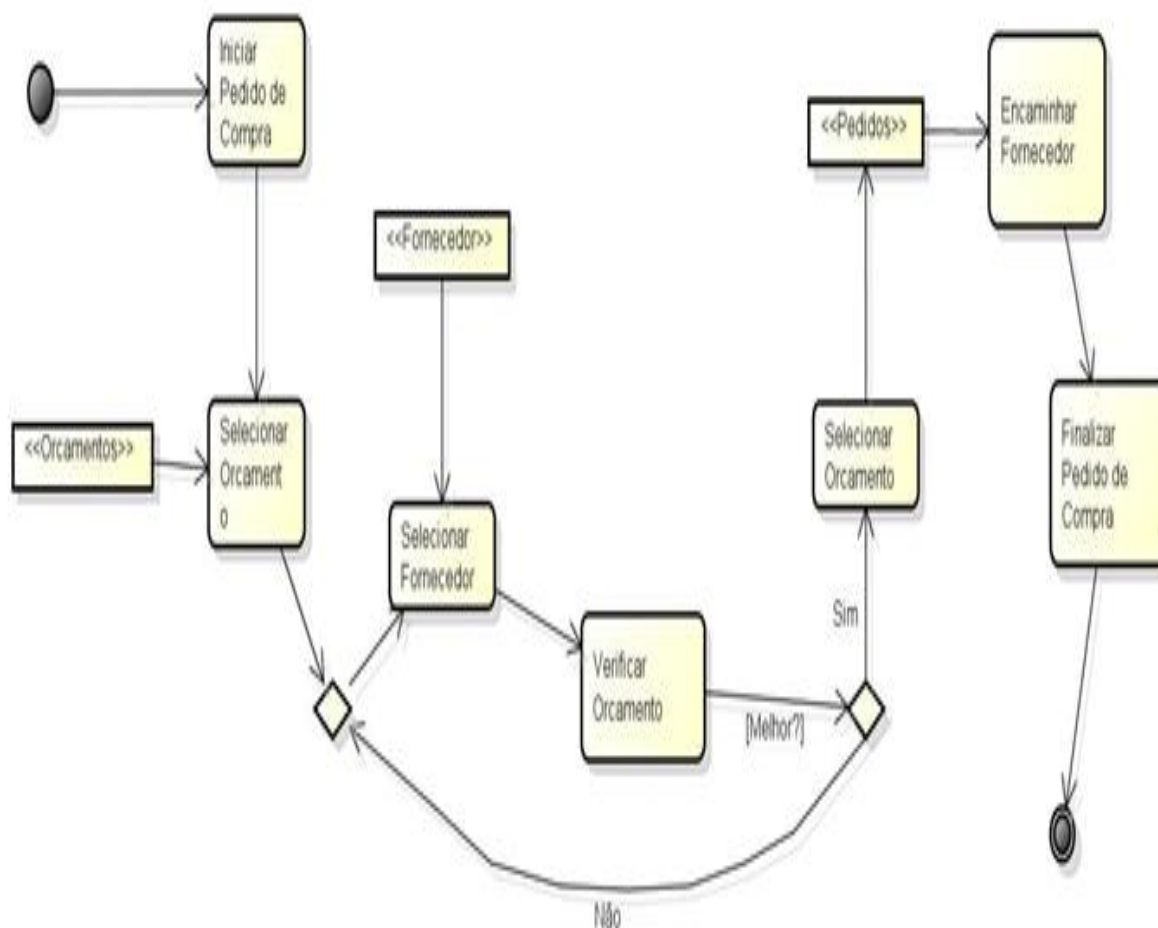


Figura 1.18 - Diagrama de Atividade
Fonte: Elaborado pelos autores, 2013.

2 ANÁLISE E PROJETO DO SISTEMA

2.1 ANÁLISES DE REQUISITOS

O escritório de advocacia para o qual foi desenvolvido o sistema de banco de dados possui um sistema que é utilizado pelos advogados para cadastrar os dados necessários para manter o cliente, e também seu tipo de processo, mas é um sistema em defasagem, que funciona em hardware desatualizado, e que não satisfaz completamente as necessidades dos advogados. Observa-se, portanto, que o escritório necessita de um novo software gestor que possua algumas características do sistema anterior.

Propõe-se então a implantação de um sistema de gestão com banco de dados para ambiente Web que satisfaça os usuários alvo para que seu trabalho seja facilitado.

2.1.1 Análise de Requisitos Funcionais

O sistema a ser implantado precisa permitir acesso por mais de um usuário ao mesmo tempo e alteração simultânea com controle de acesso concorrente, coisa que o sistema atual não é capaz de fazer.

É perceptível também que os advogados precisam ser lembrados de seus compromissos a partir de relatórios e prazos de entrada e movimentação das ações, contando além de tudo com o armazenamento e utilização de observações sobre o cliente, sentença do juiz, andamento do processo e demais necessidades que o sistema antigo não supre.

Há a necessidade de uma relação de prazos de pagamento que foi combinado com o cliente, além disso, há a necessidade de que um cliente possa ter mais de uma pasta cadastrada para ele para que assim seja possível o cadastro de mais de um tipo de ação para este cliente, pois para cada ação é utilizada uma pasta diferente.

É necessário que haja a possibilidade de cadastrar, alterar, pesquisar, e excluir cliente, réu, ação, pasta e pagamento.

Os dados a serem cadastrados do cliente são código, nome, endereço, RG, CPF, dois telefones para contato sendo um celular e um residencial.

Os dados a serem cadastrados do réu, assim como os do cliente são código, nome, RG, CPF, endereço e dois telefones para contato sendo um residencial e outro celular.

Para a ação, armazenar o número do processo, data da distribuição, data a entrar com a ação, data para próxima alteração na ação, valor da ação, data da sentença, tipo de sentença, tipo de ação, e observações.

Os campos da pasta são número da pasta, identificação do armário, identificação da gaveta do armário, há a necessidade de pesquisar o número da pasta.

Os atributos do pagamento são data do pagamento combinada com o cliente, data do recebimento, quantidade de parcelas, valor da parcela, valor total a pagar e valor total recebido até o momento.

2.2 PERSPECTIVAS DO PRODUTO

O produto mantém seu funcionamento independente, ou seja, não depende de nenhum software para suas funcionalidades. Tem o intuito de atender as necessidades contidas pelo usuário no escritório de advocacia, focando simplificar e informatizar algumas tarefas de seu trabalho cotidiano.

2.3 CARACTERÍSTICAS

Abaixo na Tabela 1 são demonstradas as necessidades e prioridades das funções do novo sistema:

Tabela 1.1 - Necessidades e Prioridades

Necessidade	Prioridade
Cadastro de Usuário	Alta
Cadastro de Pessoa	Alta
Cadastro de Processo	Alta
Cadastro de Pagamento	Alta
Cadastro de Sentença	Alta

Cadastro de Pasta	Alta
Cadastro de Cidade	Media
Relatório de Usuário	Alta
Relatório de Processo	Alta
Relatório de Cliente por Processo	Alta
Relatório de Listagem de Processos em Andamento	Alta
Relatório de Listagem de Processos Encerrados	Alta
Relatório de Clientes	Alta
Relatório de Recebimentos	Alta
Relatório de Contas a Receber	Alta
Pesquisas básicas	Media
Relatório de Pastas por Clientes	Alta

Fonte: Elaborado pelos autores 2013.

2.4 DIAGRAMA DE CASO DE USO

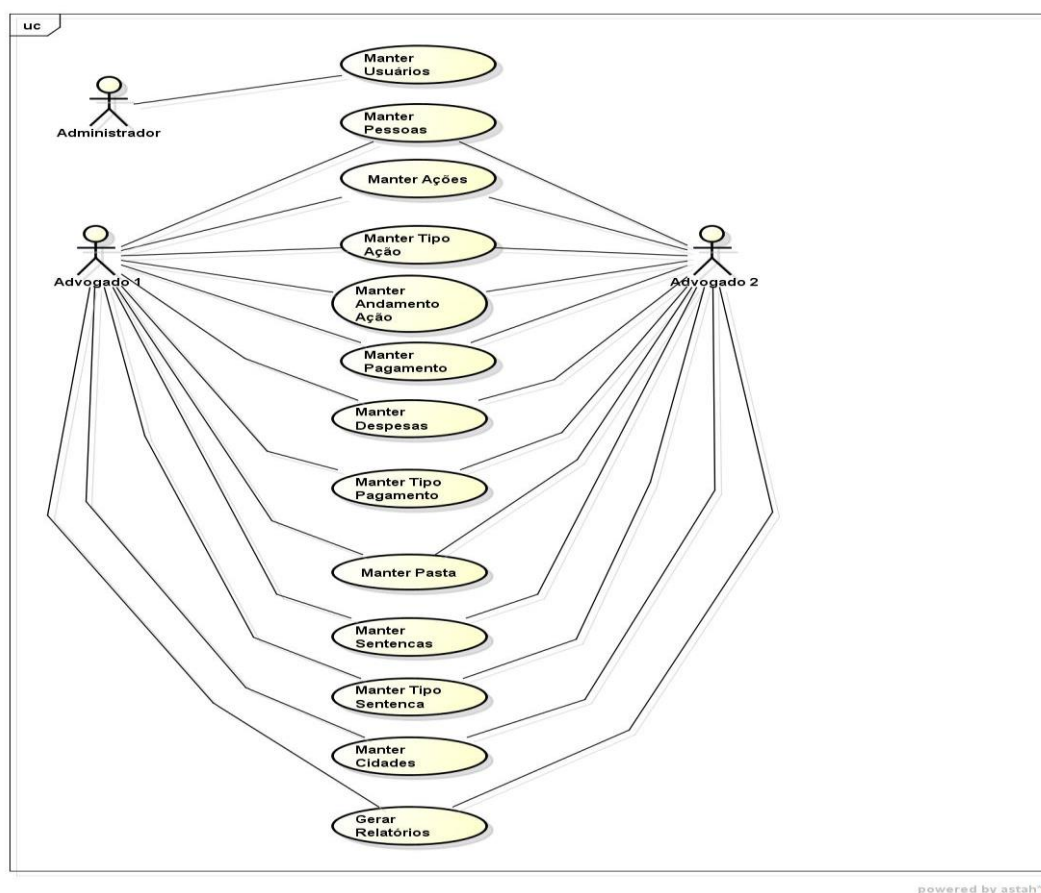


Figura 2.1 - Diagrama de Caso de Uso

Fonte: Elaborado pelos autores, 2013.

2.4.1 Especificação de Histórias

- 1- Como advogado eu gostaria de realizar o cadastro dos meus clientes.
- 2- Como advogado eu gostaria de alterar os dados dos clientes que estão cadastrados no sistema.
- 3- Como advogado eu gostaria de pesquisar o número da pasta de acordo com o nome do cliente.
- 4- Como advogado eu gostaria de poder excluir um ou mais clientes cadastrados.
- 5- Como advogado eu gostaria de pesquisar os dados dos meus clientes fornecendo o nome do cliente ao programa.
- 7- Como advogado eu gostaria de realizar o cadastro de réus.
- 8- Como advogado eu gostaria de alterar os dados dos réus cadastrados.
- 9- Como advogado eu gostaria de excluir um ou mais réus cadastrados.
- 10- Como advogado eu gostaria de pesquisar os dados dos réus cadastrados fornecendo o nome destes ao programa.
- 11- Como advogado eu gostaria de cadastrar ação e seu tipo e os clientes que entraram com esta ação, e também incluir réu e interessados na ação.
- 12- Como advogado eu gostaria de poder excluir ações cadastradas.
- 13- Como advogado eu gostaria de poder alterar os dados das ações cadastradas.
- 14- Como advogado eu gostaria de pesquisar qual ação é do cliente que eu forneci o nome ao programa.
- 15- Como advogado eu gostaria de cadastrar o tipo de pagamento que pode ser por enquanto cartão, cheque ou dinheiro.
- 16- Como advogado eu gostaria de poder alterar a forma de pagamento.
- 17- Como advogado eu gostaria de manter as pastas dos meus clientes podendo cadastrar mais de uma pasta para cada cliente.
- 18- Como administrador eu gostaria de cadastrar usuários para o sistema.
- 19- Como administrador eu gostaria de alterar usuários cadastrados no sistema.
- 20- Como administrador eu gostaria de pesquisar usuários no sistema fornecendo o nome.
- 21- Como administrador eu gostaria de poder excluir usuários do sistema.

2.4.2 Análise de Requisitos Não Funcionais

- A interface com o usuário deve ser amigável, isso será possível com o uso de JSF, PrimeFaces, HTML com CSS o que facilita criação de interfaces Web amigáveis. Prioridade importante.
- Usuários simultâneos, permitir e controlar acesso concorrente à base de dados. Prioridade essencial.

2.5 ANÁLISE E DESIGN

2.5.1 Projeto do Banco de dados

2.5.1.1 Diagrama de entidade e relacionamento

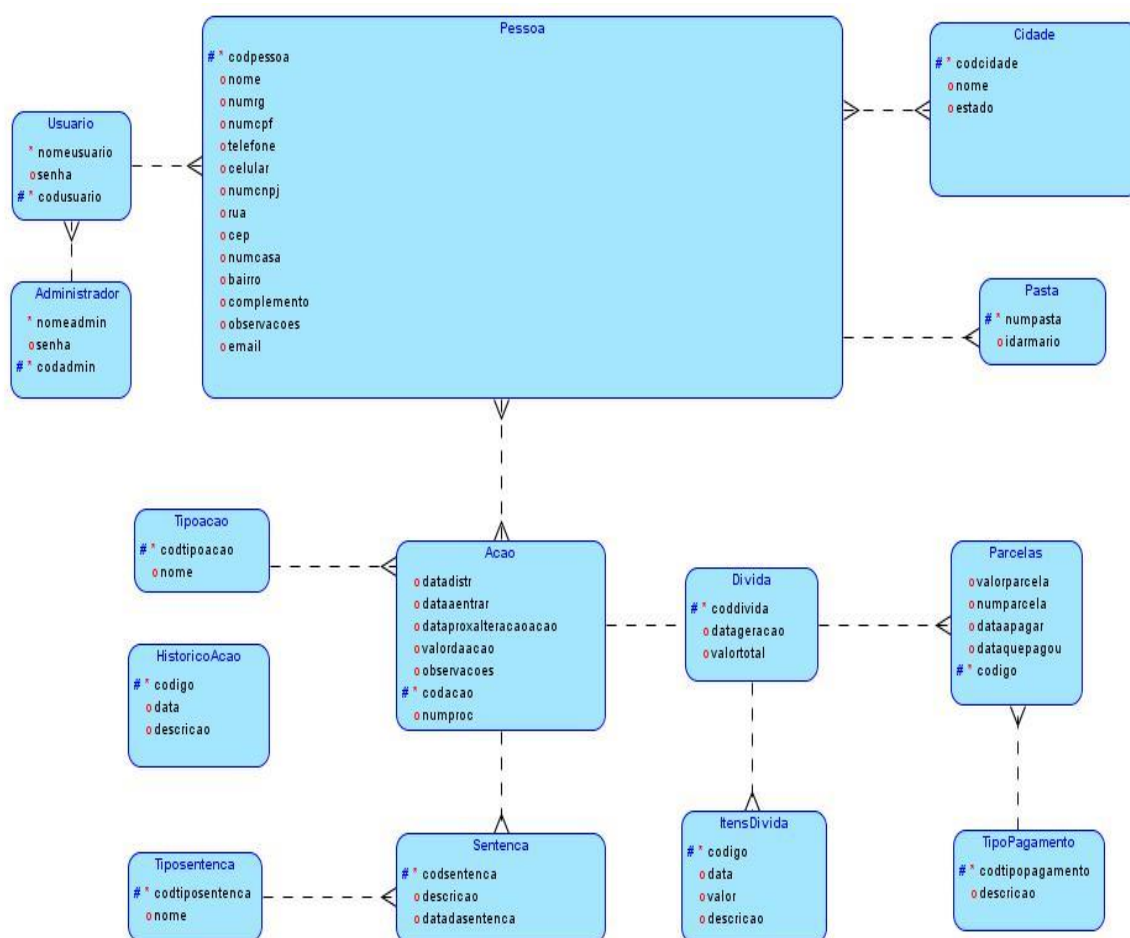


Figura 2.2 - Diagrama de Entidade e Relacionamento (Modelo Lógico) do Sistema

Fonte: Elaborado pelos autores 2013.

2.6 DIAGRAMA DE CLASSE

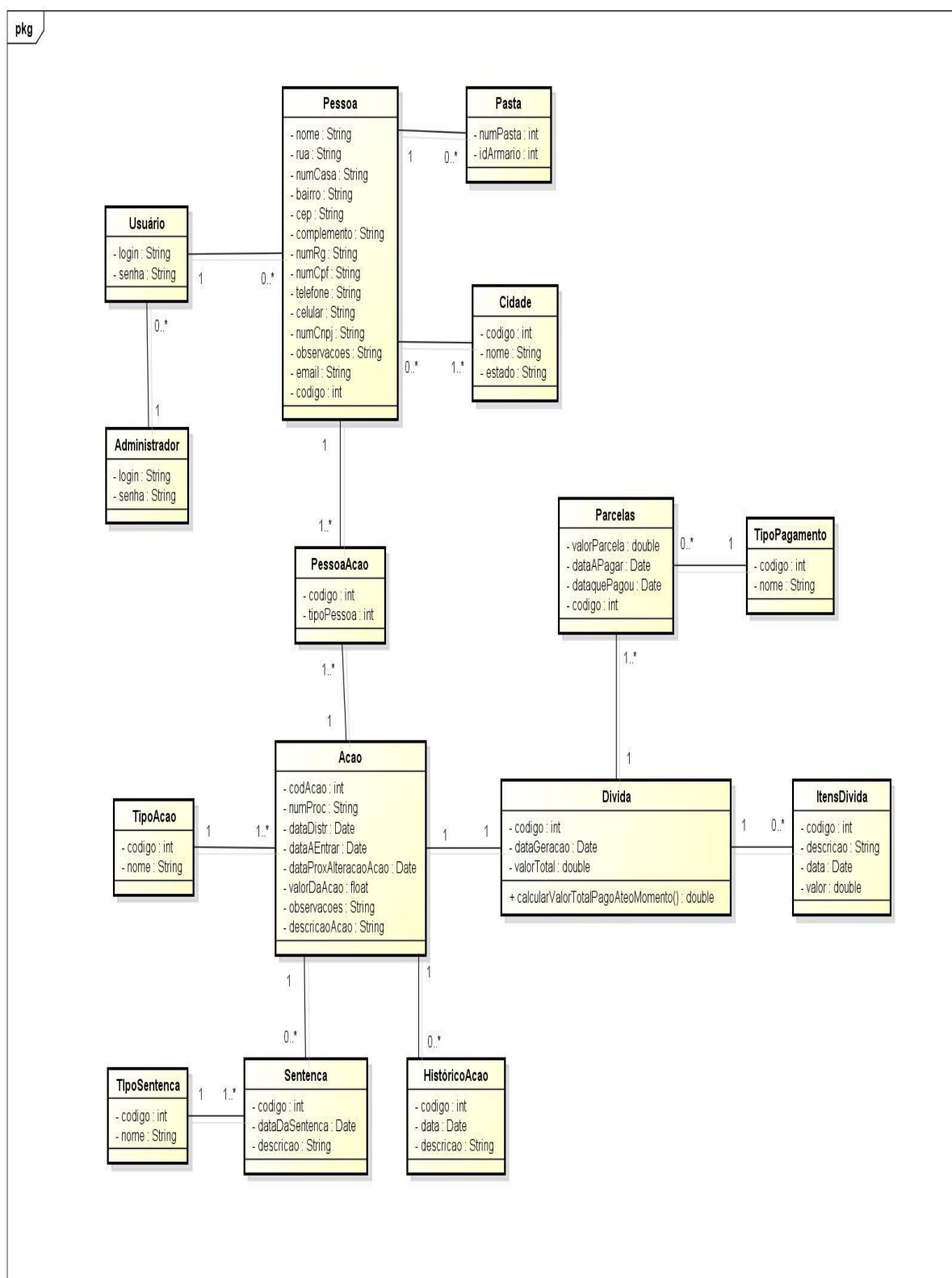


Figura 2.3 - Diagrama de Classe do Sistema
Fonte: Elaborado pelos autores 2013.

2.7 DIAGRAMAS DE ATIVIDADES

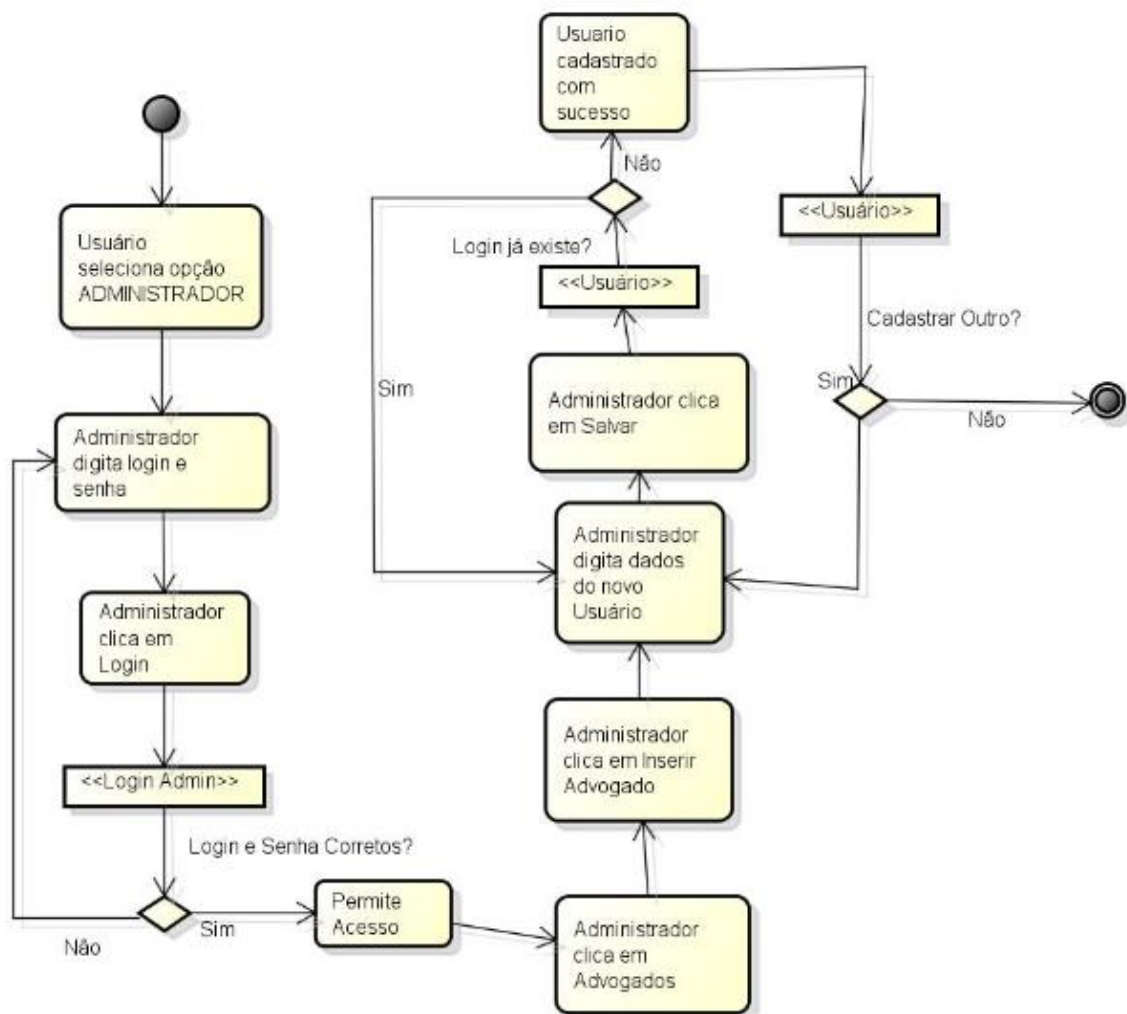


Figura 2.4 - Diagrama de Atividade Cadastro de Usuário
Fonte: Elaborado pelos autores 2013.

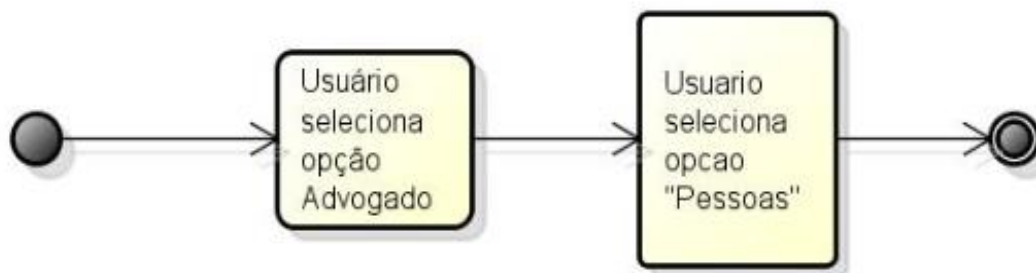


Figura 2.5 - Diagrama de Atividade Listar Todas as Pessoas
Fonte: Elaborado pelos autores 2013.

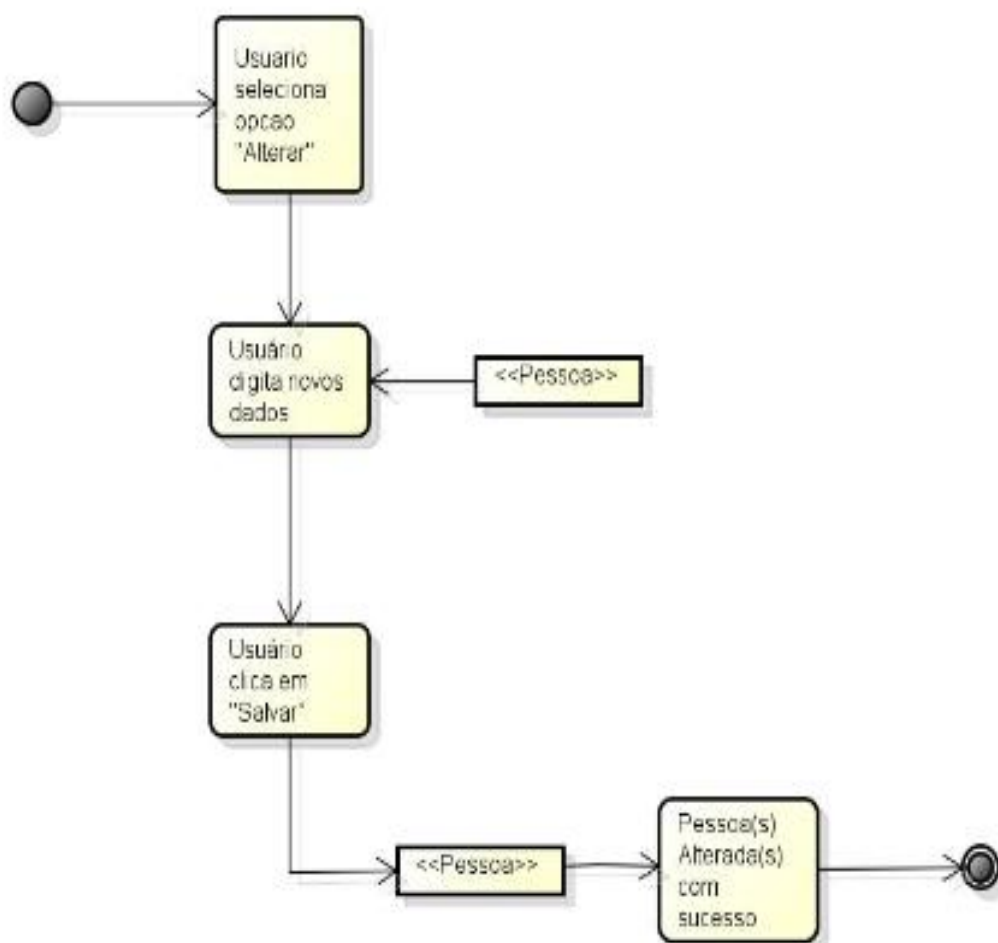


Figura 2.6 - Diagrama Atividade Alterar Pessoa
 Fonte: Elaborado pelos autores 2013.

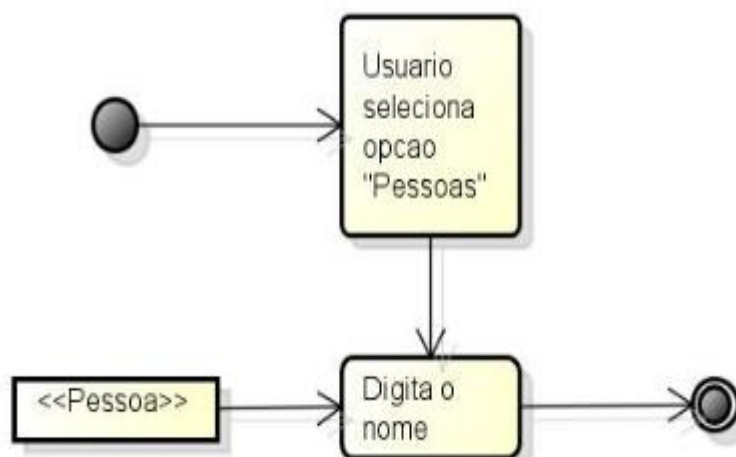


Figura 2.7 - Diagrama de Atividade Buscar Por Nome
 Fonte: Elaborado pelos autores 2013.

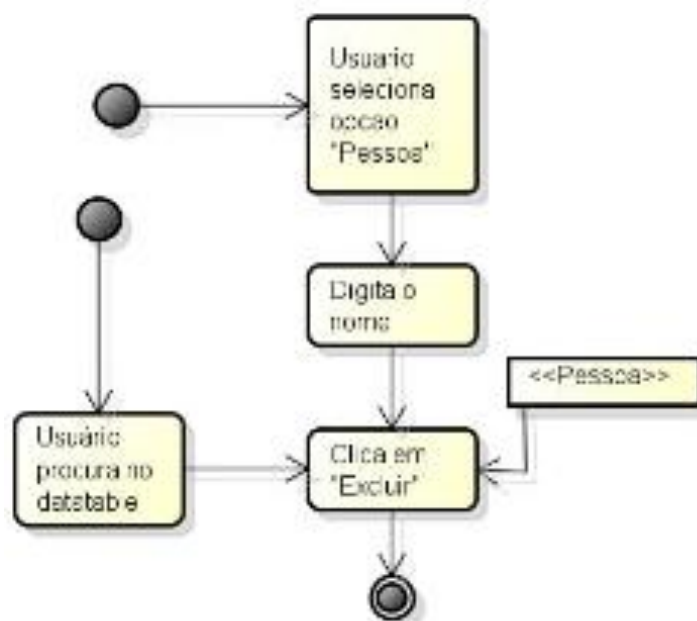


Figura 2.8 - Diagrama de Atividade Excluir Pessoa
 Fonte: Elaborado pelos autores 2013.

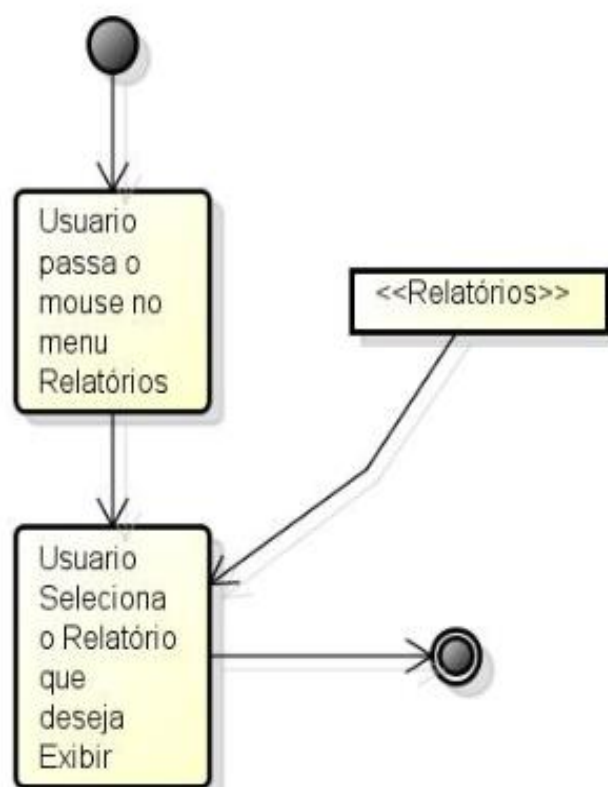


Figura 2.9 - Diagrama de Atividade Exibir Relatório
 Fonte: Elaborado pelos autores 2013.

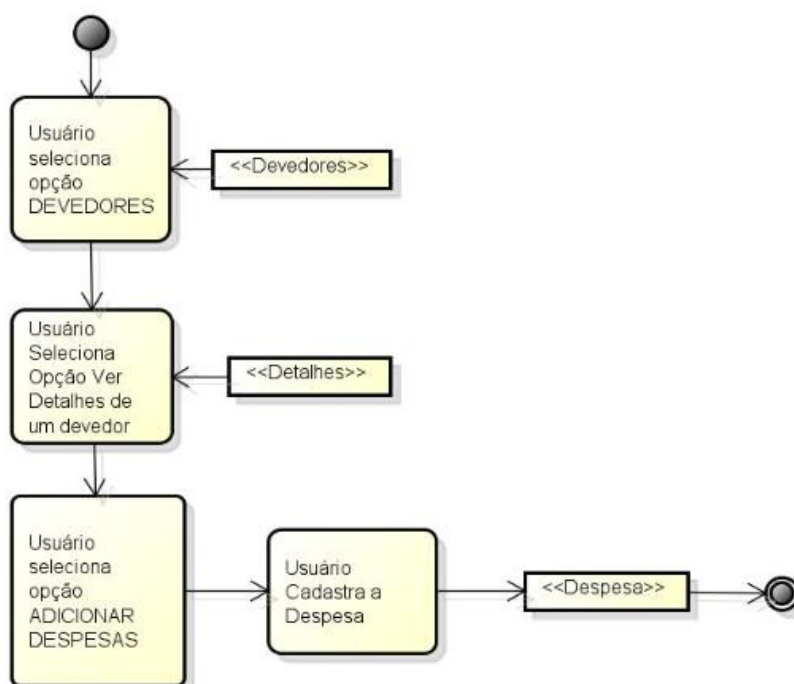


Figura 2.10 - Diagrama de atividades Gerar Despesas
Fonte: Elaborado pelos autores 2014.

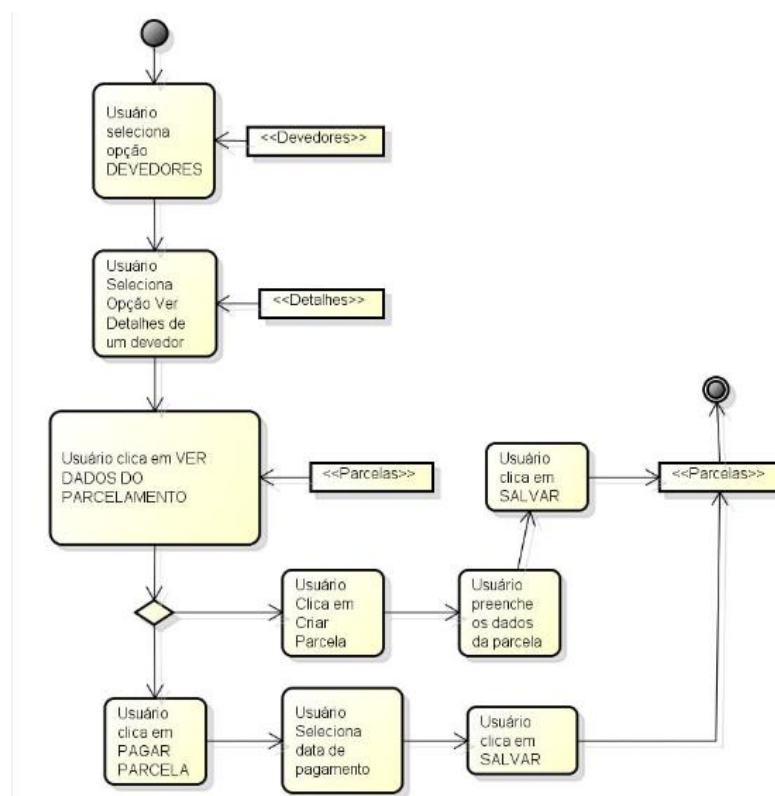


Figura 2.11 - Diagrama de Atividades Parcelamento
Fonte: Elaborado pelos autores 2014.

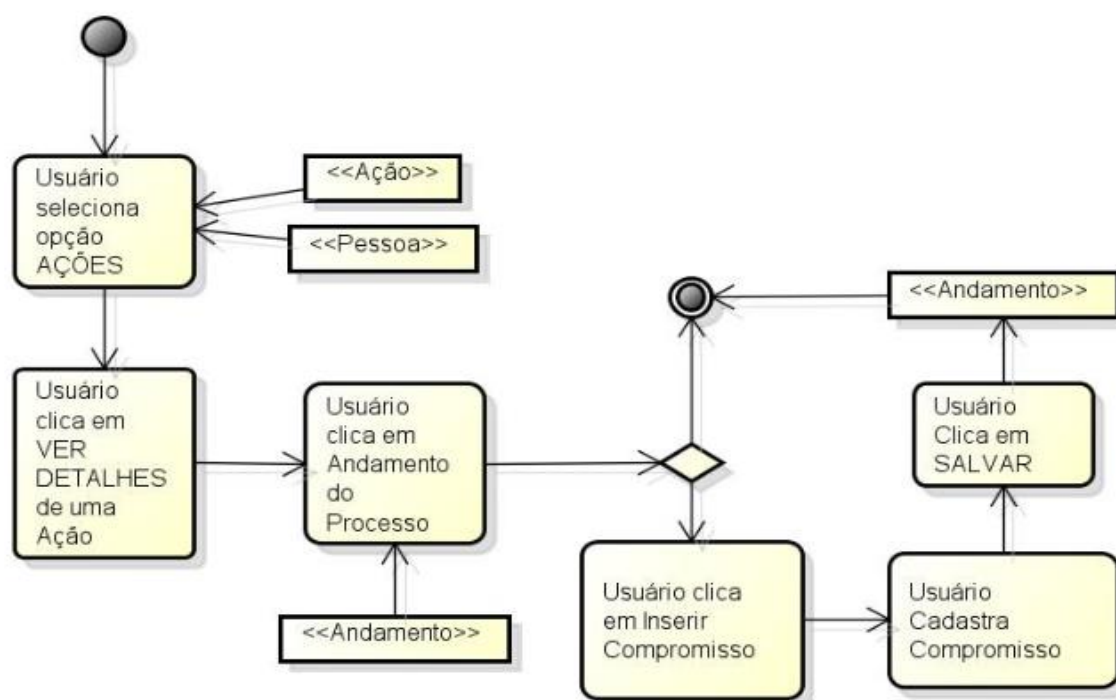


Figura 2.12 - Diagrama de Atividade Andamento de Processos
 Fonte: Elaborado pelos autores 2014.

2.8 DIAGRAMAS DE MVC

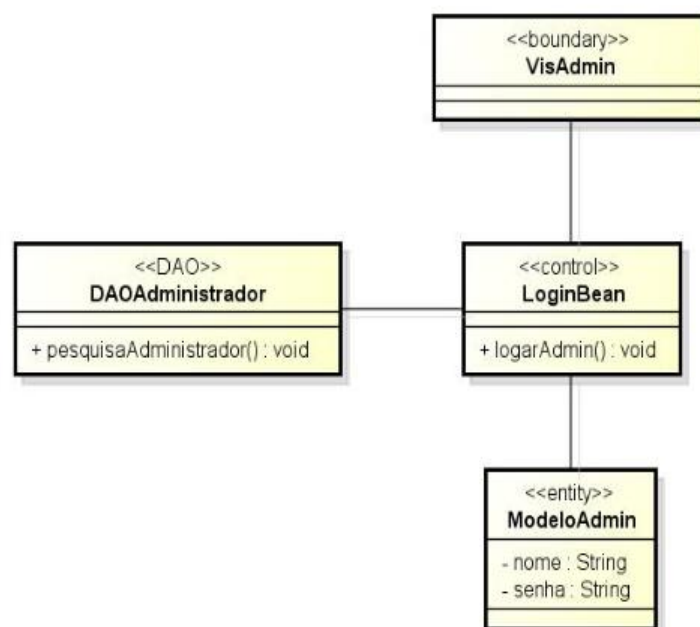


Figura 2.13 - Diagrama de MVC Login do Administrador
 Fonte: Elaborado pelos autores 2013.

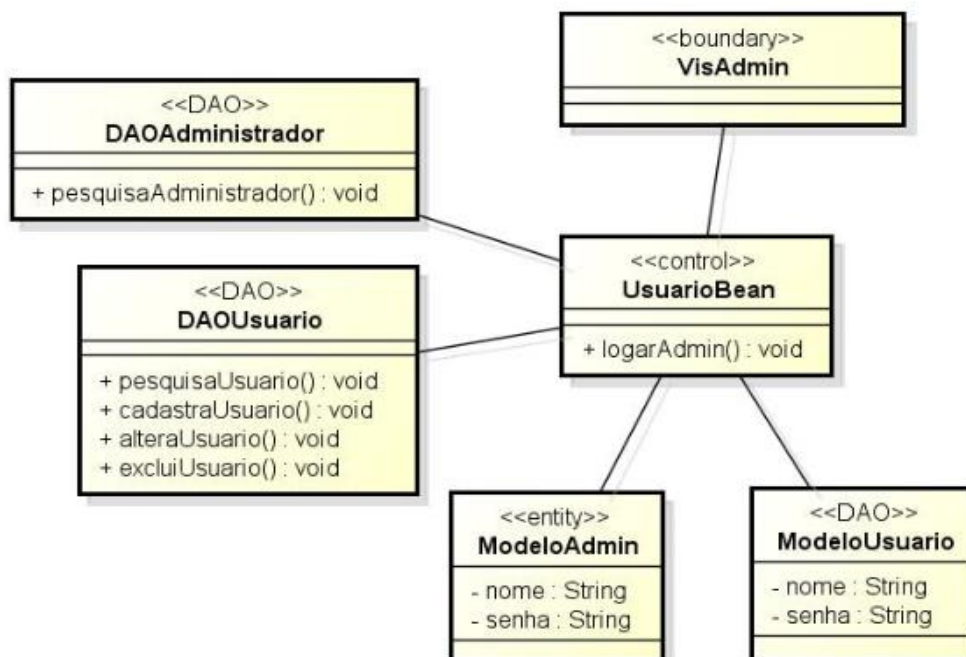


Figura 2.14 - Diagrama MVC Cadastro Usuário

Fonte: Elaborado pelos autores 2013.

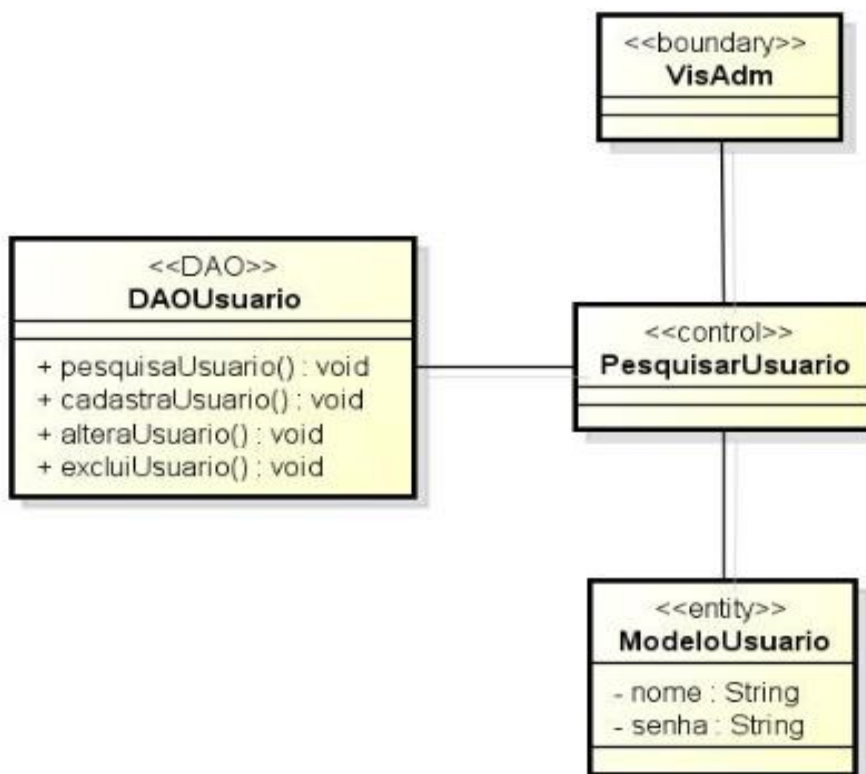


Figura 2.15 - Diagrama MVC Pesquisar Usuário

Fonte: Elaborado pelos autores 2013.

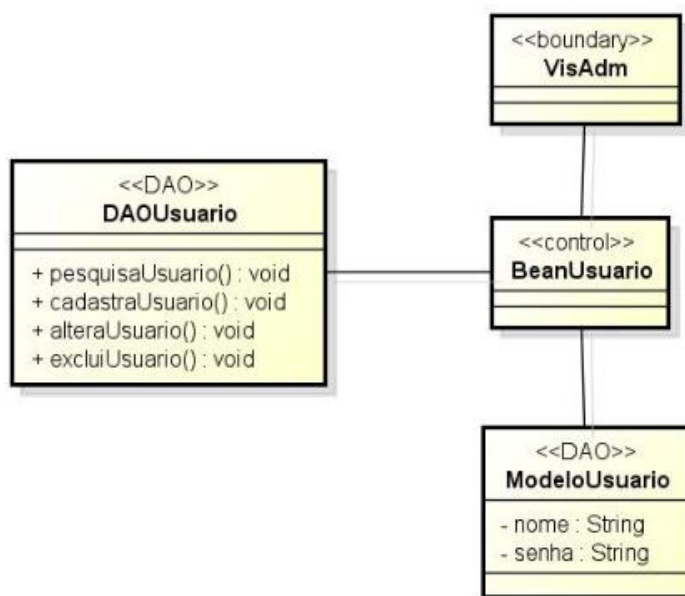


Figura 2.16 - Diagrama MVC Alterar Usuário
Fonte: Elaborado pelos autores 2013.

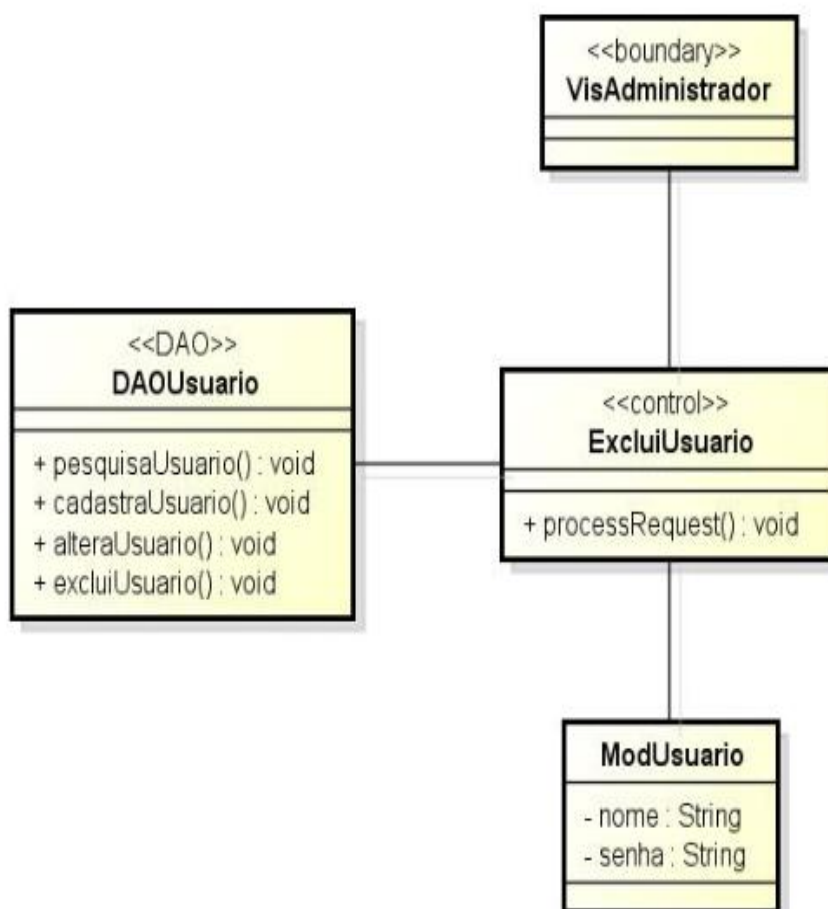


Figura 2.17 - Diagrama MVC Excluir Usuário
Fonte: Elaborado pelos autores 2013.

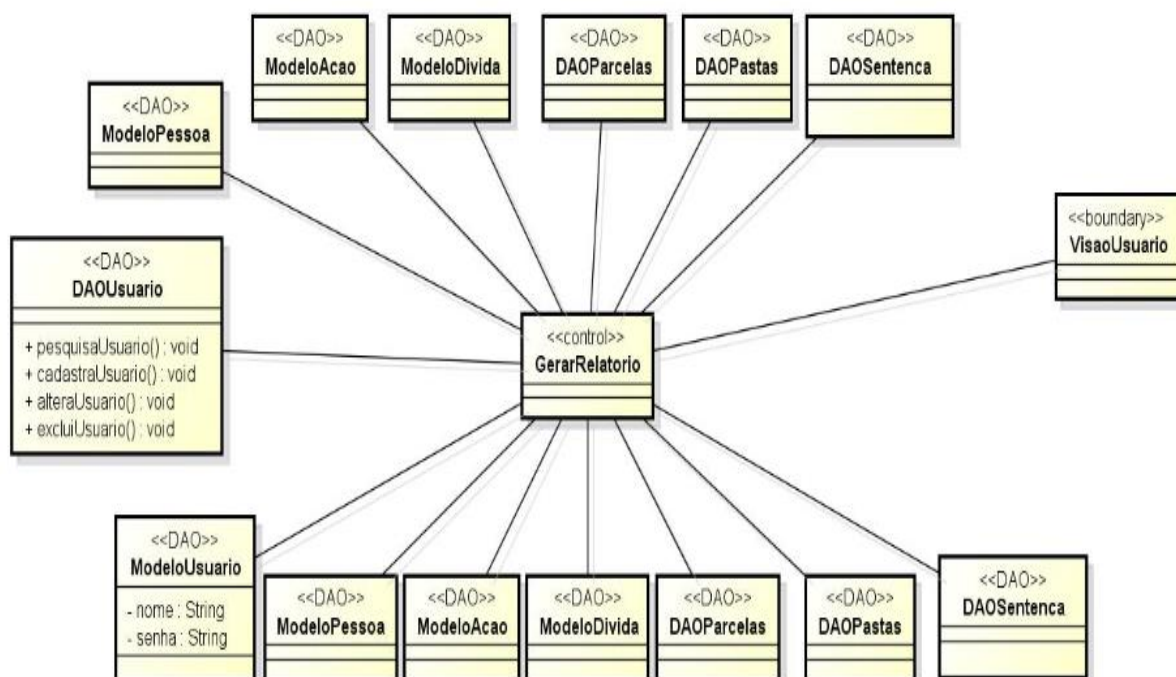


Figura 2.18 - Diagrama MVC Gerar Relatório
Fonte: Elaborado pelos autores 2013.

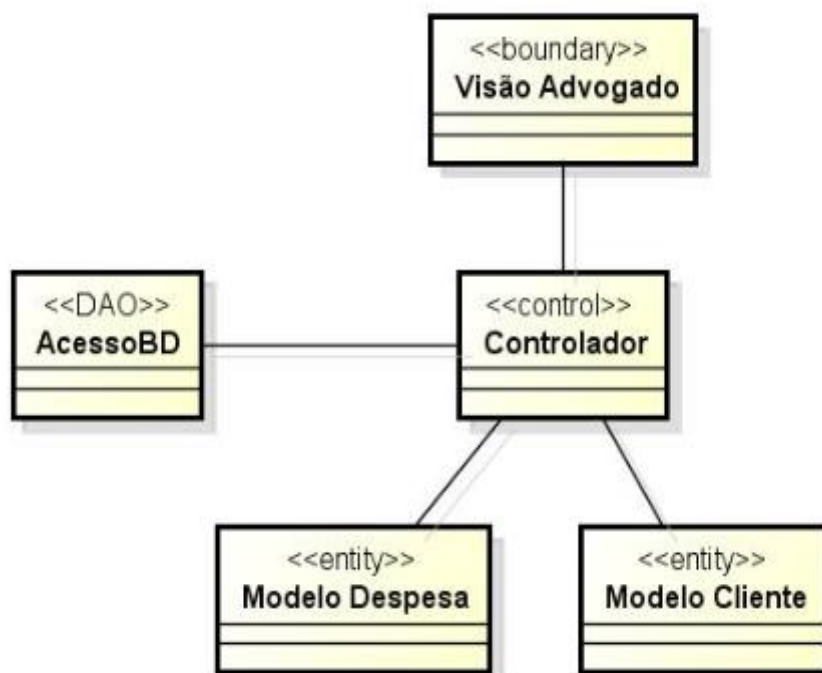


Figura 2.19 - MVC Gerar Despesa
Fonte: Elaborado pelos autores 2014.

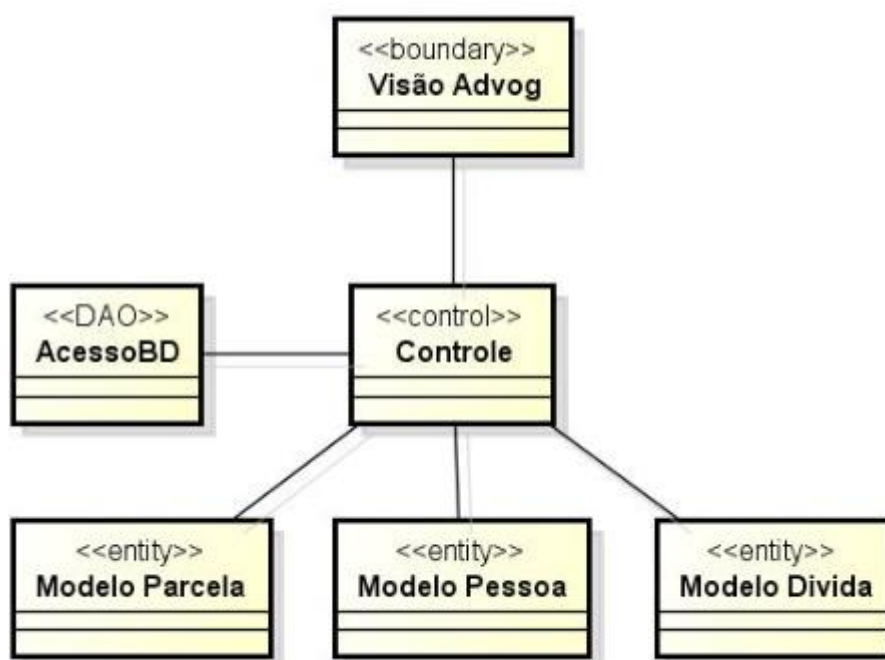


Figura 2.20 - MVC Parcelamento
 Fonte: Elaborado pelos autores 2014.

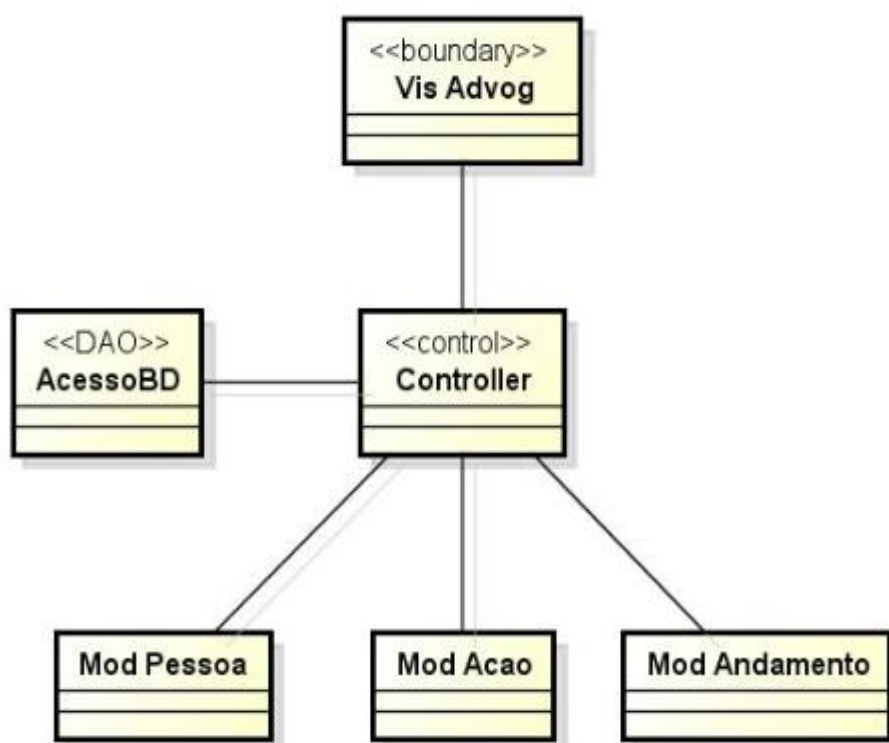


Figura 2.21 - MVC Andamento do Processo
 Fonte: Elaborado pelos autores 2014.

3.9 DIAGRAMAS DE SEQUÊNCIA

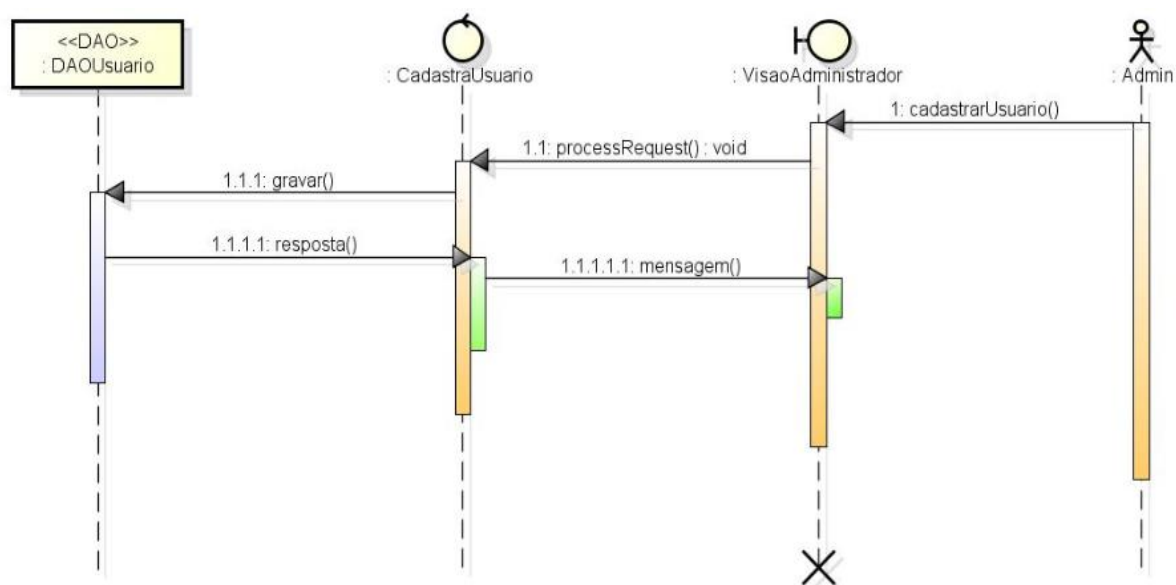


Figura 2.22 - Diagrama de Sequência Cadastrar Usuário

Fonte: Elaborado pelos autores 2013.

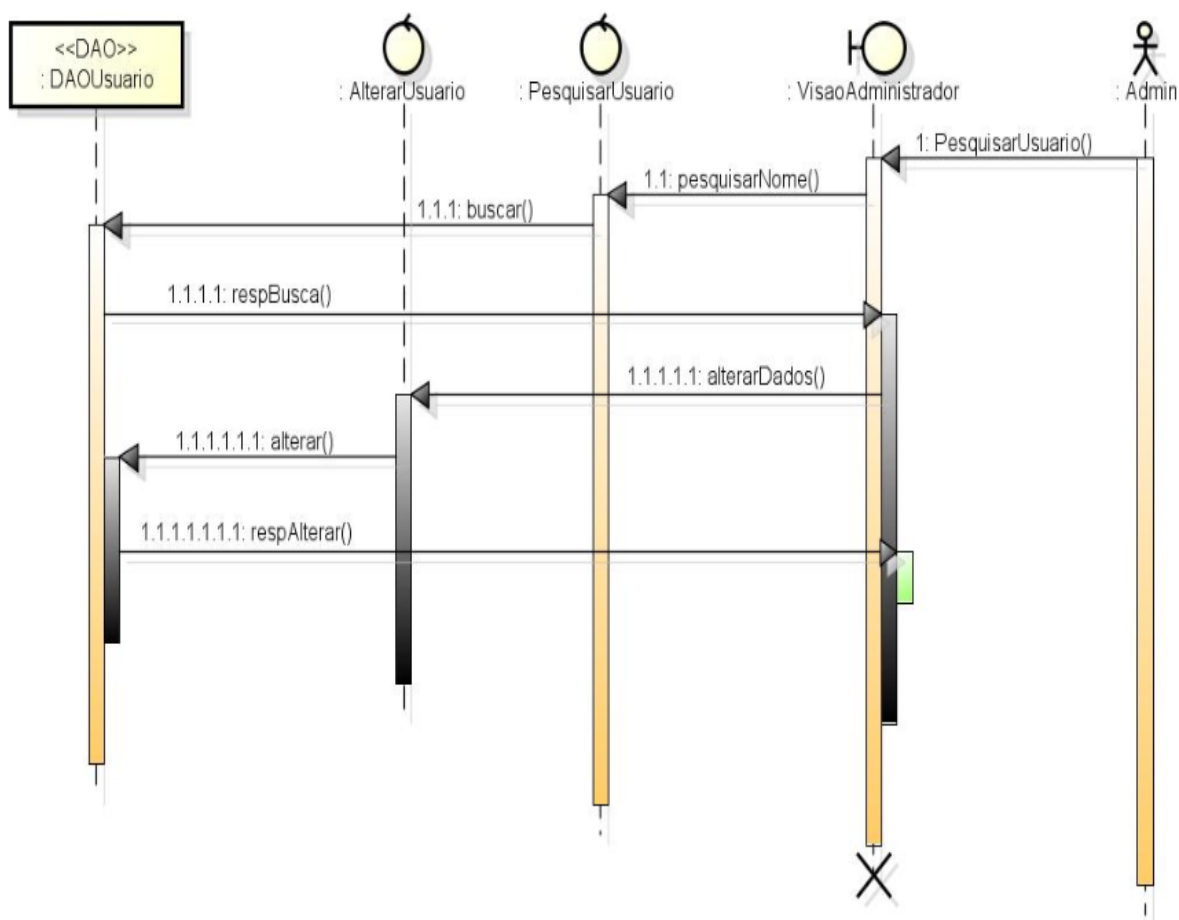


Figura 2.23 - Diagrama de Sequência Alterar Usuário

Fonte: Elaborado pelos autores 2013.

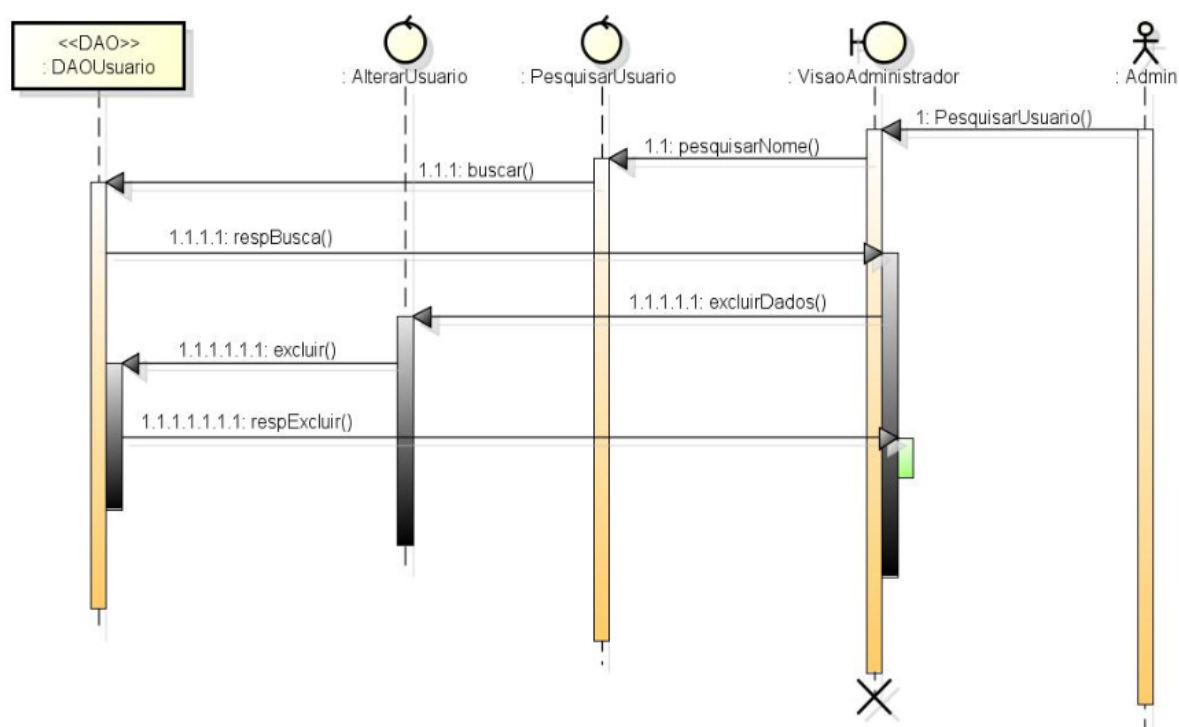


Figura 2.24 - Diagrama de Sequência Excluir Usuário
Fonte: Elaborado pelos autores 2013.

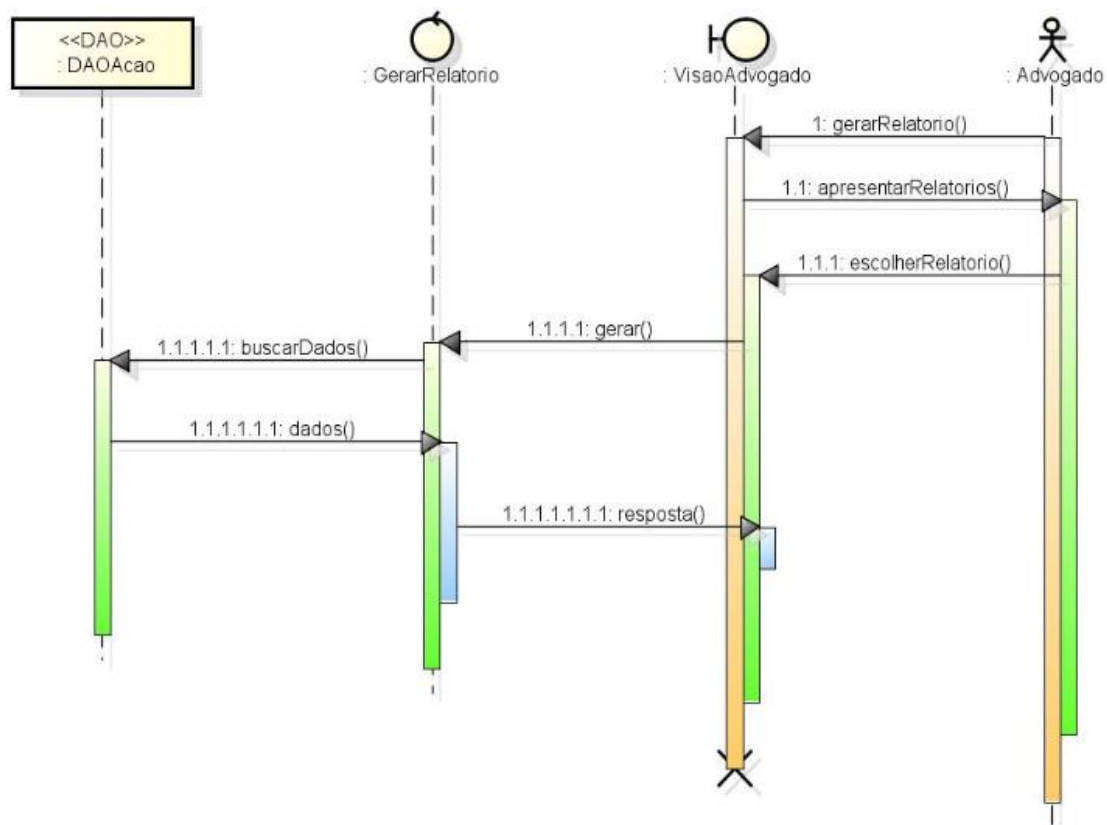


Figura 2.25 - Diagrama de Sequência Gerar Relatório
Fonte: Elaborado pelos autores 2013.

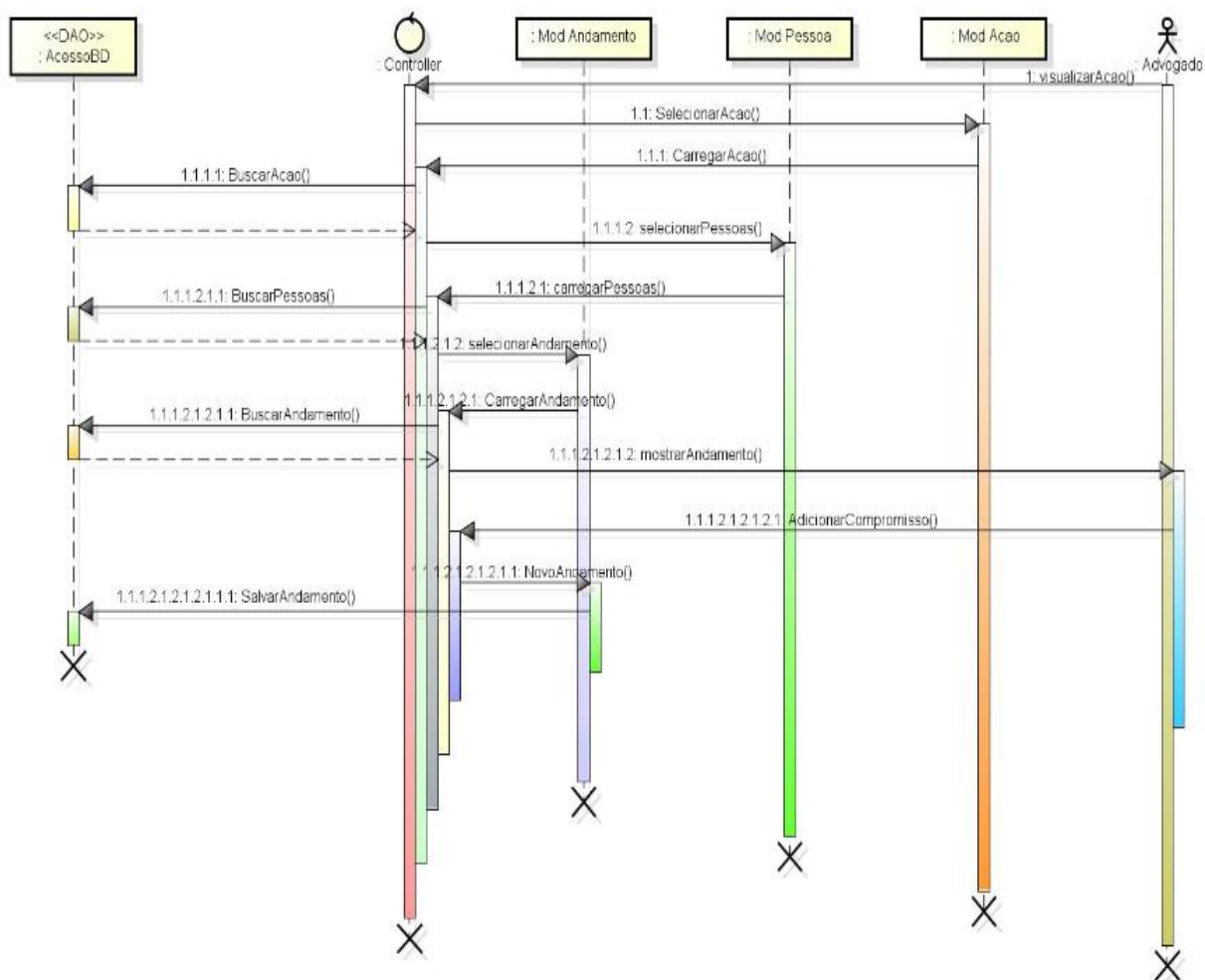


Figura 2.26 - Diagrama de Sequência do Andamento de Processo
Fonte: Elaborado pelos autores 2014.

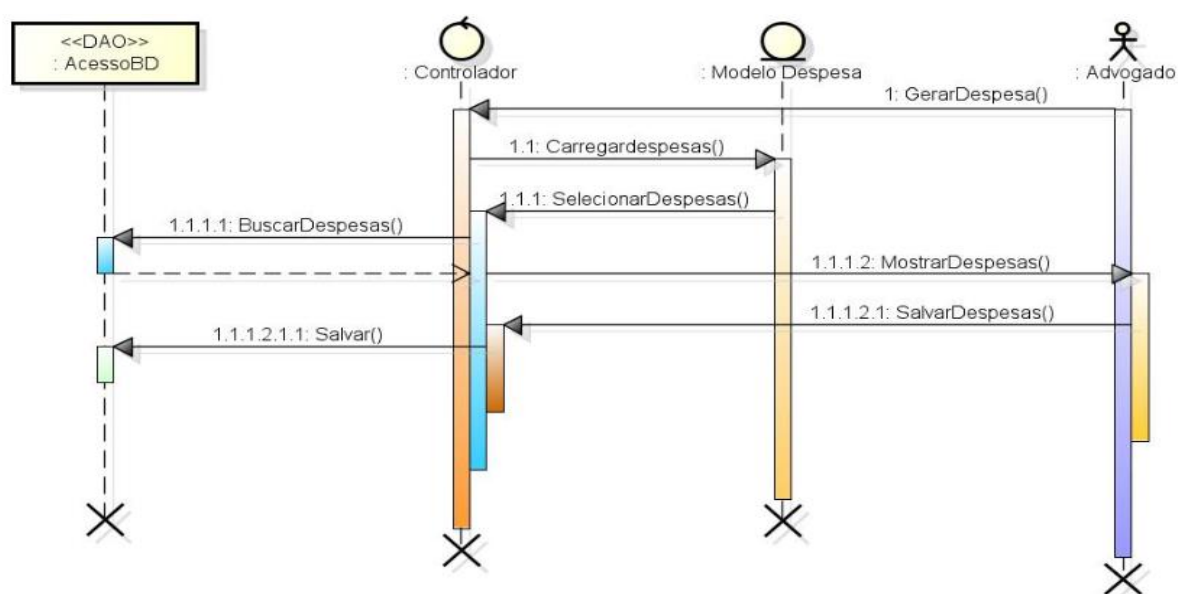


Figura 2.27 - Diagrama de Sequência Gerar Despesas
Fonte: Elaborado pelos autores 2014.

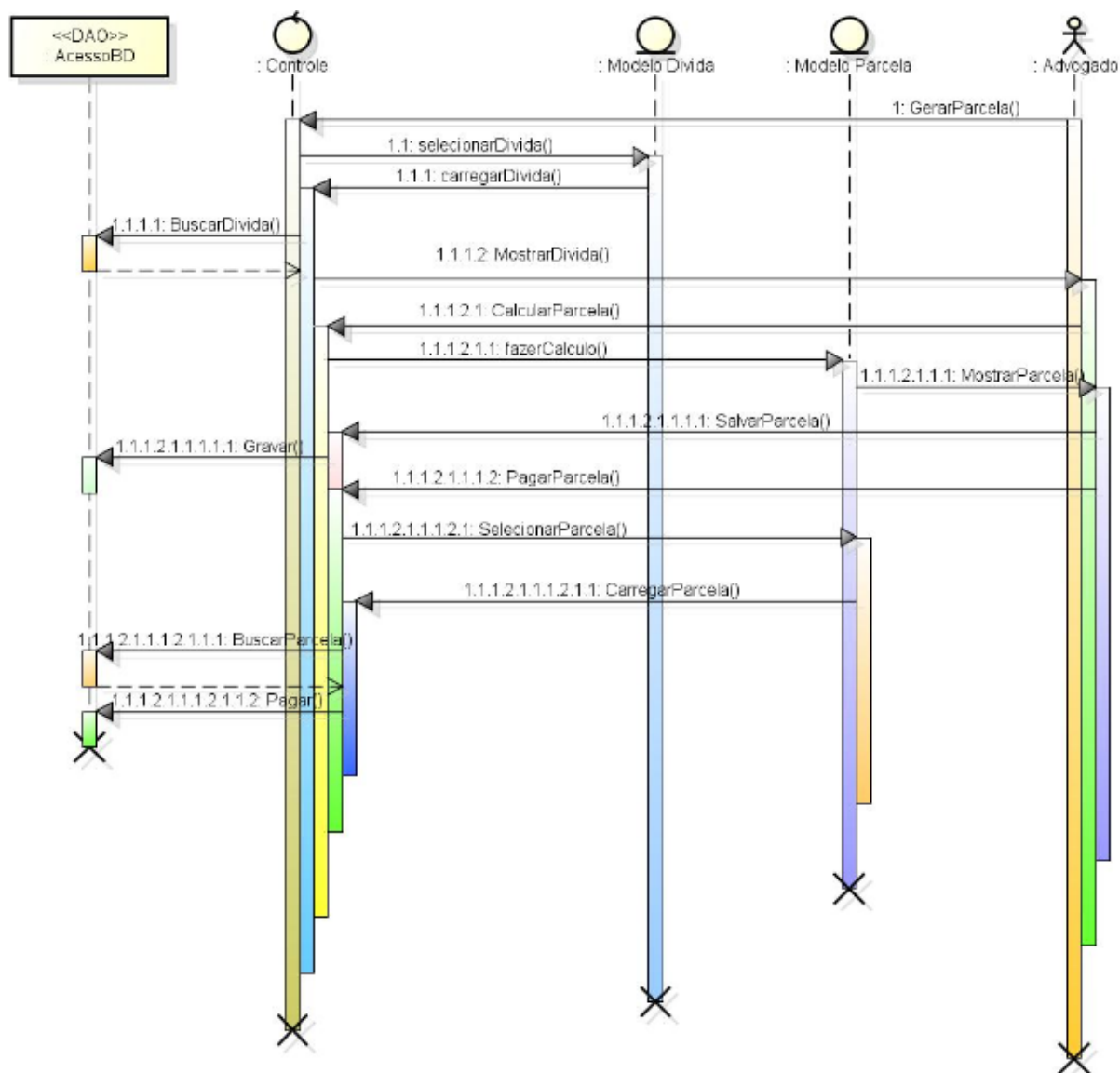


Figura 2.28 - Diagrama de Sequência gerar e pagar parcelas
 Fonte: Elaborado pelos autores 2014.

3 DISCUSSÕES DOS RESULTADOS



Figura 3.1 - Página Inicial do Sistema
Fonte: Elaborado pelos autores 2014.

Na figura 3.1, a tela inicial do sistema, nesta página tem-se um cabeçalho onde está descrito o nome do sistema e o nome dos proprietários da empresa. Já na lateral existe um menu com duas opções para os módulos Administrador e Advogado. Lembrando que o administrador já vem cadastrado com um nome e senha previamente.

Destaque para a utilização de tags *include* para o cabeçalho, menu, rodapé e *insert* para o conteúdo.

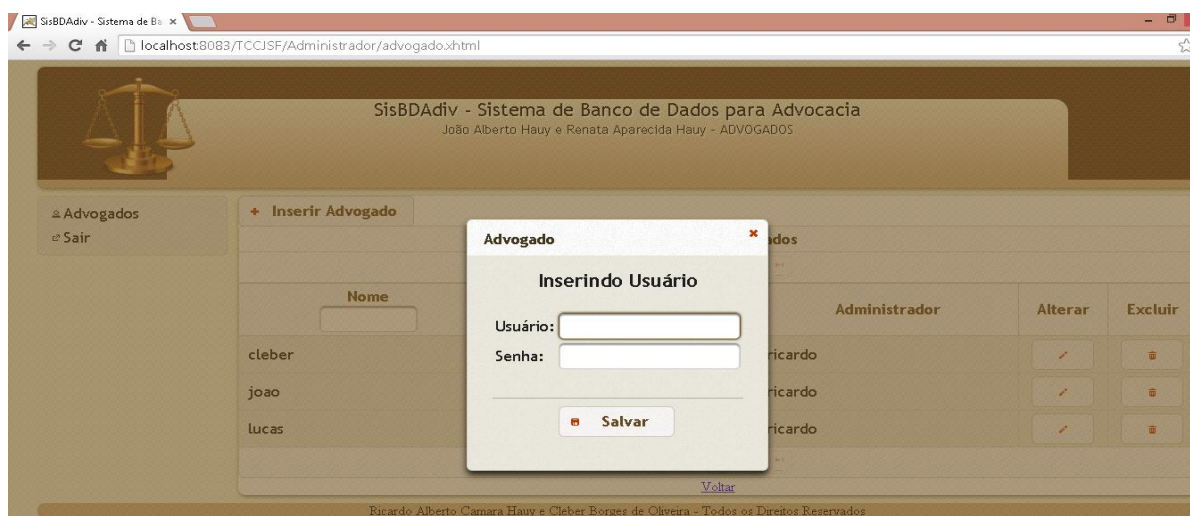


Figura 3.2 – Tela de Cadastro de Advogados para Acessar o Sistema
Fonte: Elaborado pelos autores 2014.

Na figura 3.2 mostra-se o módulo administrador na tela onde o administrador cadastra um novo usuário (advogado) para ter acesso ao sistema em seu módulo. Os campos do cadastro são usuário e senha, campos obrigatórios para que se tenha acesso ao sistema e não haja falhas de segurança.

Destaque para o elemento *separator*, essa linha que separa os campos do botão e para estilização CSS para manter os campos alinhados corretamente.



Figura 3.3 - Tela de Listagem dos Usuários do Sistema
Fonte: Elaborado pelos autores 2014.

Na figura 3.3, a tela de usuários (advogados) já cadastrados no sistema, essa tela é visualizada apenas pelo administrador do sistema. Nesta tela temos os botões onde é possível fazer uma alteração de usuário e senha e até mesmo a exclusão de usuário. Também temos o campo nome onde podemos fazer uma busca de usuário (advogado) por nome.

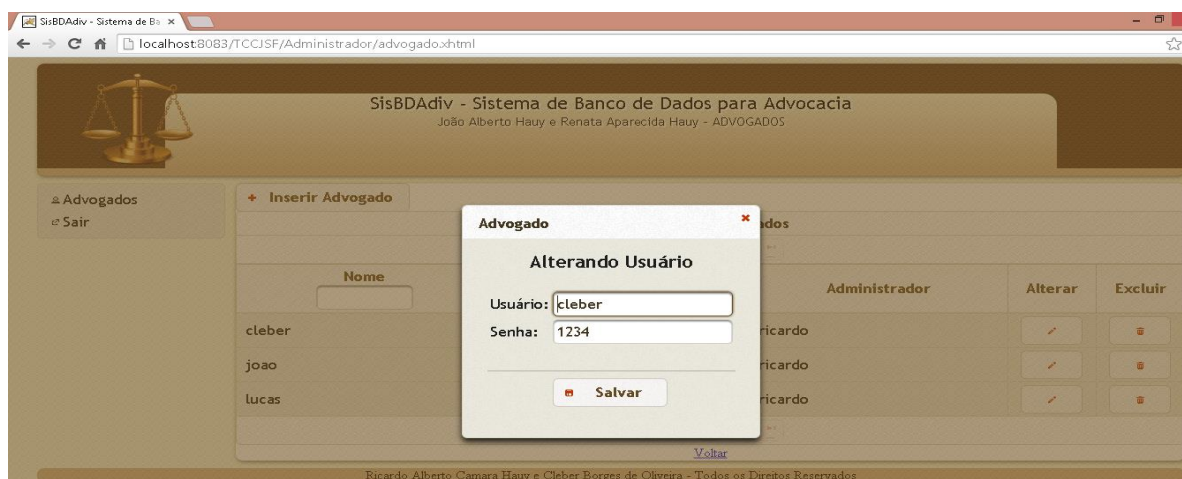


Figura 3.4 - Tela de Alteração de Usuário pelo Administrador
Fonte: Elaborado pelos autores 2014.

Na figura 3.4 é destacada a tela de alteração de usuário (advogado), que esta disponível apenas no modulo administrador, onde o administrador pode fazer alterações de usuário ou senha à partir de um clique no botão alterar.



Figura 3.5 - Tela do Módulo Advogado

Fonte: Elaborado pelos autores 2014.

Na figura 3.5 pode-se observar o módulo advogado, o usuário logado, conforme destaca a notificação no lado direito superior da tela, já no lado esquerdo da tela, pode ser visto, na horizontal, o menu, onde temos as funcionalidades do sistema disponíveis ao usuário.

Destaque para o elemento *graphicImage* utilizado para inserção das imagens.

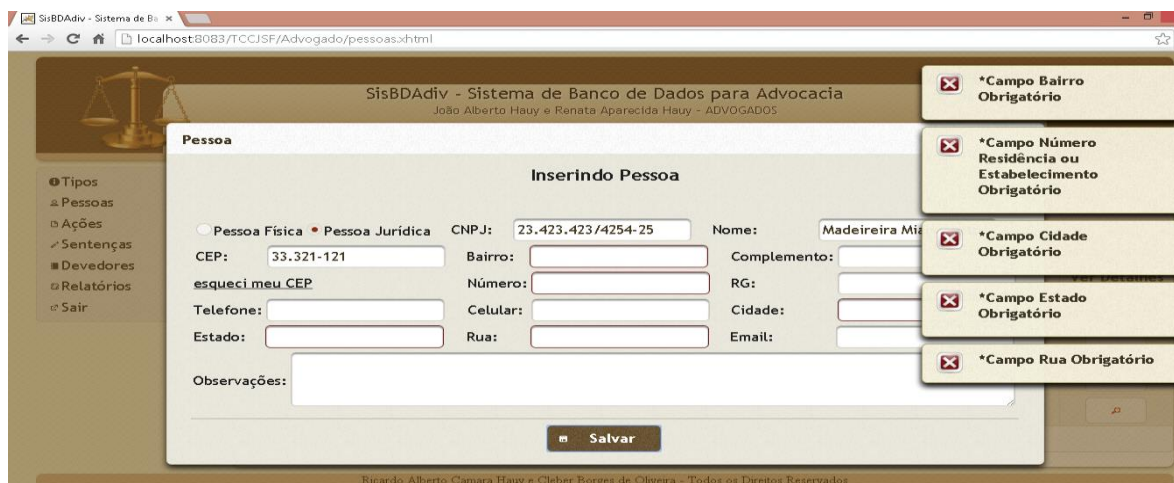


Figura 3.6 – Tela de Cadastro de Pessoas Envolvidas nas Ações

Fonte: Elaborado pelos autores 2014.

Na figura 3.6 é mostrada a tela de cadastro de pessoas envolvidas nas ações, que pode ser do tipo pessoa física ou jurídica. Os campos deste cadastro são obrigatórios, pois é onde se mantem o cadastro de clientes, réus e interessados na ação.

Destaque para as mensagens de campo obrigatório que são tags de campos com o atributo *required* igual a *true*.

SisBDAdiv - Sistema de Banco de Dados para Advocacia
João Alberto Haury e Renata Aparecida Haury - ADVOGADOS

Pessoa

Alterando Pessoa

☒ Pessoa Física ☐ Pessoa Jurídica

CPF: 123.421.431-23 Nome: Joao

CEP: 39.339-893 Bairro: Centro Complemento: Casa 2

esqueci meu CEP Número: 3 RG: 33.333.333-3

Telefone: (33) 3333-3333 Celular: (22) 99878-6353 Cidade: LINS

Estado: SP Rua: Marmoré Email: joao@1234.com.br

Observações: Data de Nascimento = 18/12/1989
Estado Civil = Solteiro

Salvar

Ricardo Alberto Camara Haury e Cleber Borges de Oliveira - Todos os Direitos Reservados

Figura 3.7 - Tela Alterando Pessoa

Fonte: Elaborado pelos autores 2014.

A Figura 3.7 é a demonstração da funcionalidade alterar pessoa que se justifica para modificação de dados já cadastrados anteriormente de maneira equivocada, nesta funcionalidade há uma junção de três classes para que seja armazenado corretamente a cidade onde a pessoa mora atualmente.

SisBDAdiv - Sistema de Banco de Dados para Advocacia
João Alberto Haury e Renata Aparecida Haury - ADVOGADOS

Tipos das Ações Cadastradas

+ Inserir Novo Tipo

Código Tipo	Tipo	Alterar	Excluir
13	Cível		
14	Criminal		
15	Trabalhista		

Volcar

Ricardo Alberto Camara Haury e Cleber Borges de Oliveira - Todos os Direitos Reservados

Figura 3.8 - Tela Tipo de Ações Cadastradas

Fonte: Elaborado pelos autores 2014.

Na Figura 3.8 tem-se a demonstração da tela na qual são listados todos os tipos de ações possíveis cadastrados que futuramente quando o usuário (advogado) for cadastrar a ação poderá ser selecionado um destes tipos.

Destaque para os botões de excluir que estão setados como *disabled* por que já existe alguma ação cadastrada destes tipos e para manter a integridade e não gerar *bugs* estão impossibilitados de serem excluídos. As telas tipo de pagamento e tipo de sentença são semelhantes a esta.

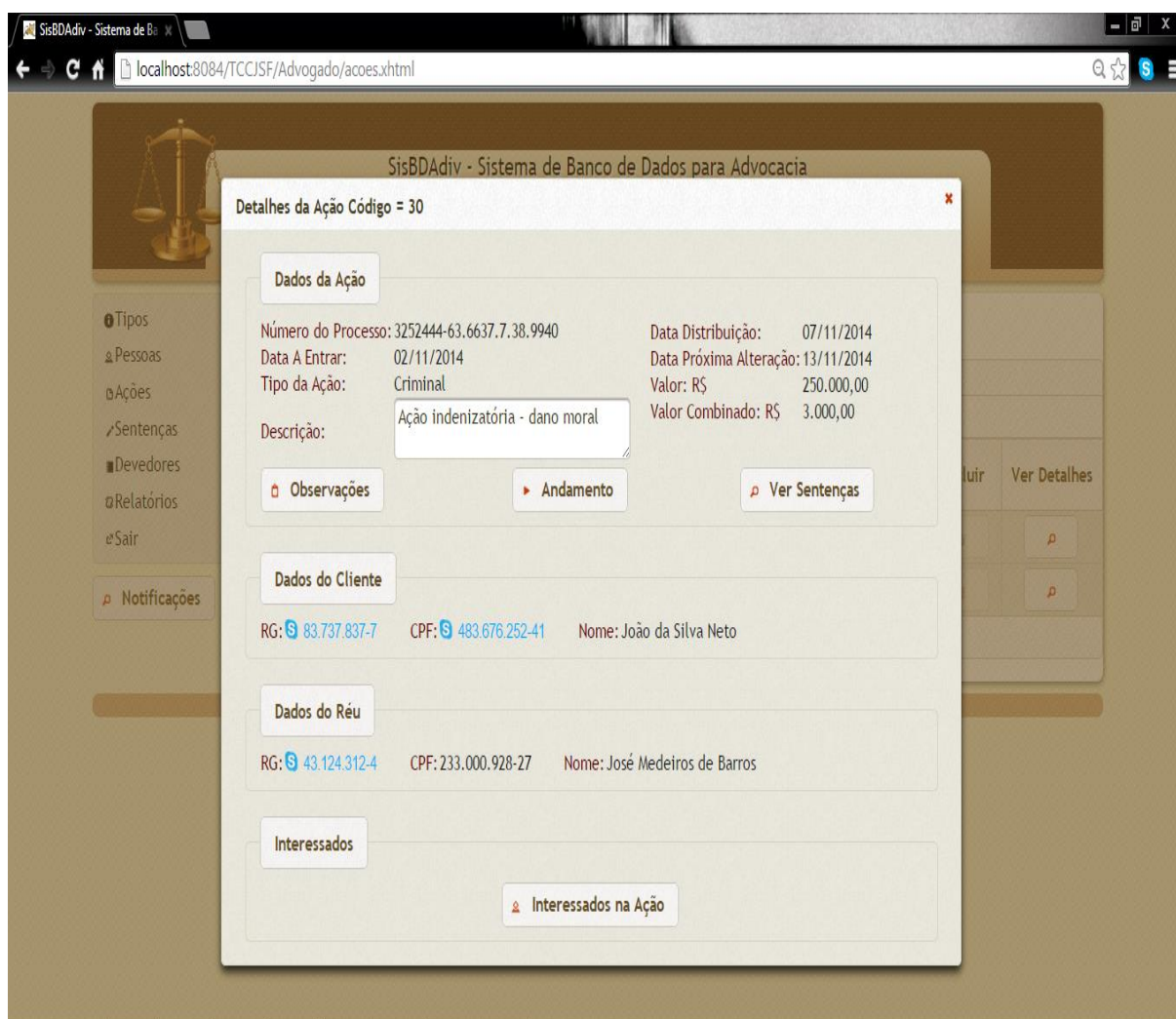


Figura 3.9 - Tela Detalhes da Ação

Fonte: Elaborado pelos autores 2014.

Os detalhes de uma ação específica juntamente com os botões que levam às observações, andamento processual e sentenças que foram dadas na ação são esboçados na figura 3.9. Nesta tela pode-se visualizar também quem são as pessoas que estão interessadas nesta ação a partir do botão Interessados na Ação.

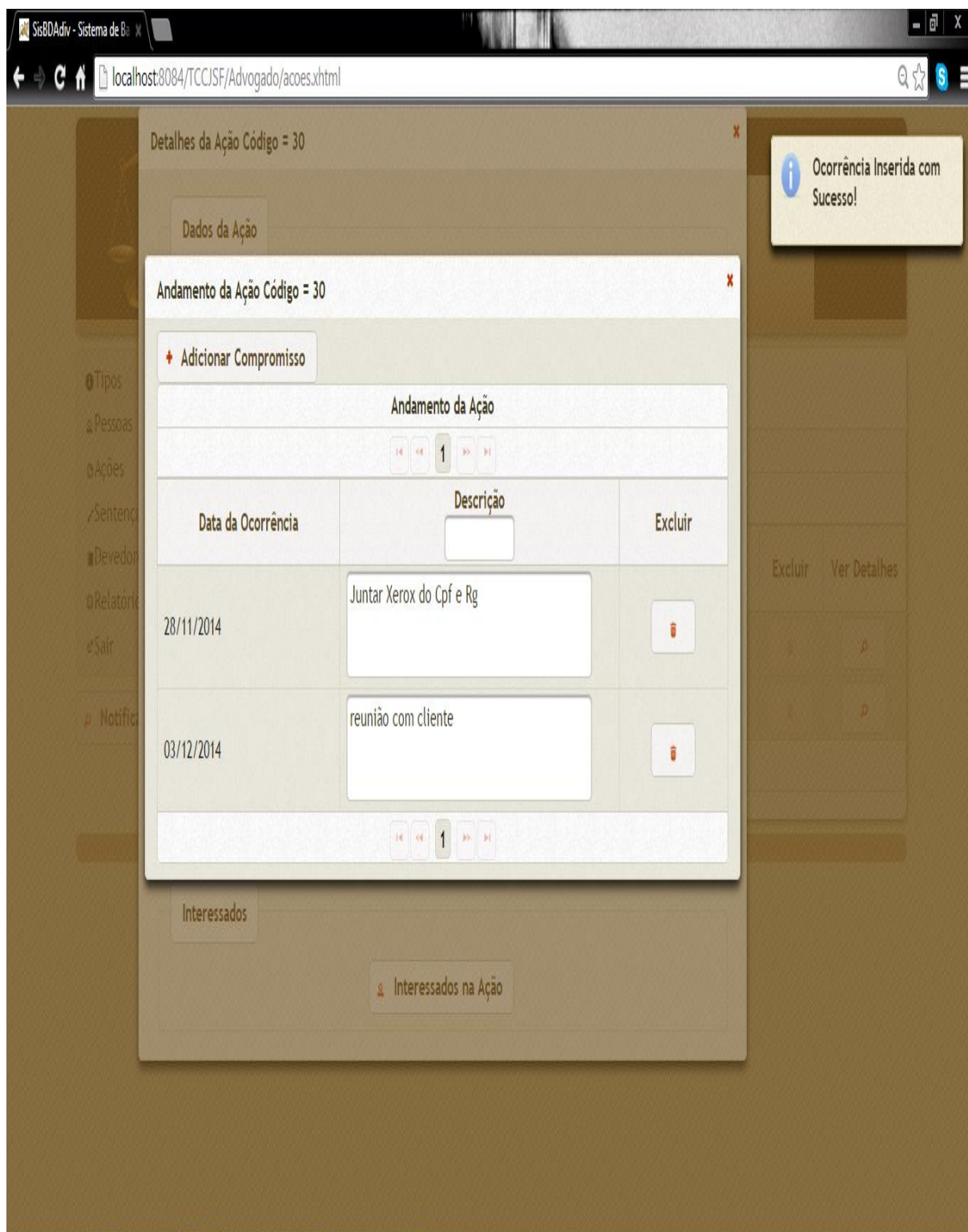


Figura 3.10 - Tela de Andamento do Processo

Fonte: Elaborado pelos autores 2014.

Na Figura 3.10 é mostrada a tela de andamento processual com as ocorrências que haverá na ação, ocorrências estas que são cadastradas pelo

usuário (advogado) e são listadas para que este possa se organizar com relação a uma ação específica.

Destaque para a utilização do componente do Primefaces *dataTable* para a listagem dos andamentos.

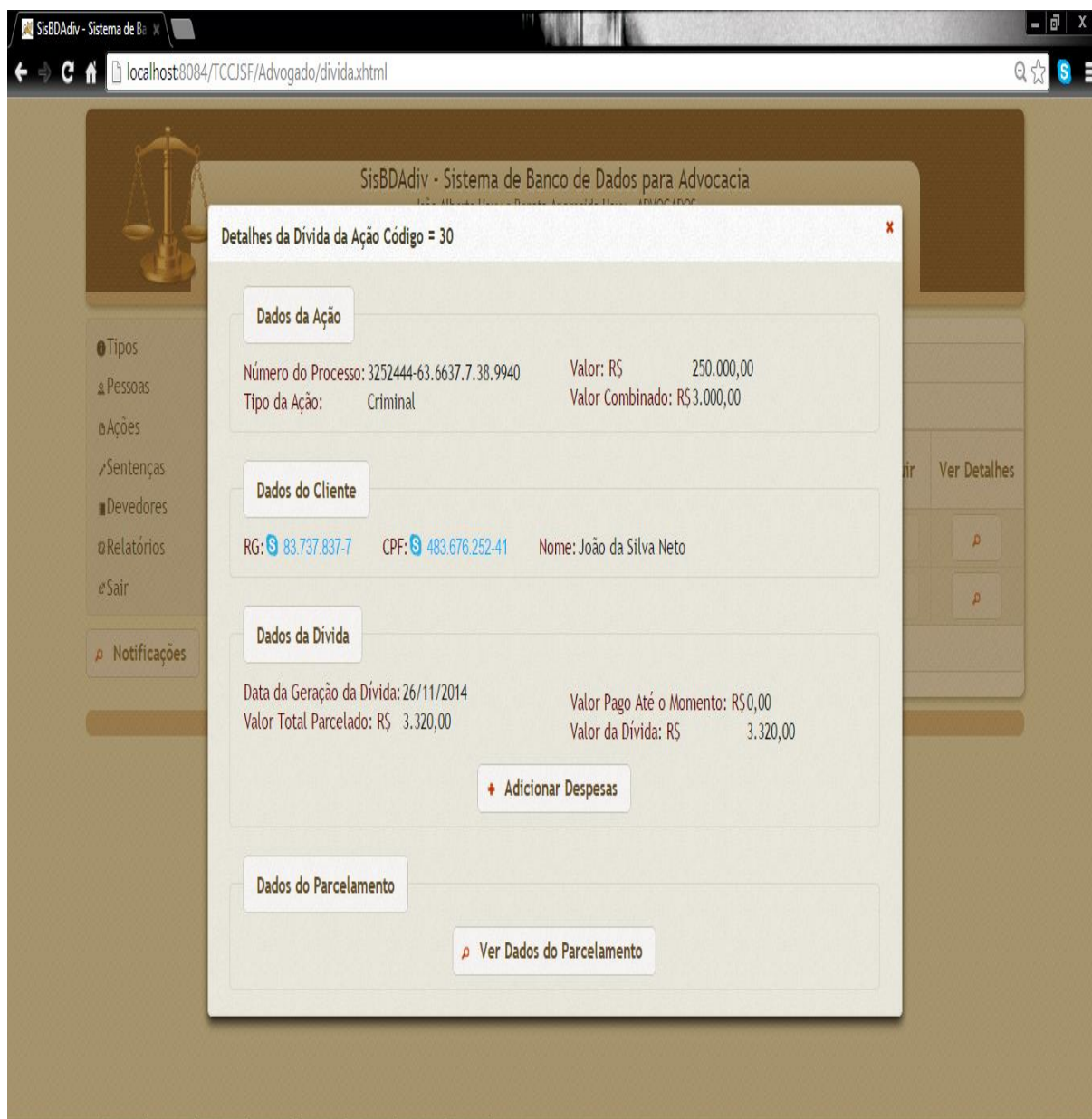


Figura 3.11 - Tela de Detalhes da Dívida

Fonte: Elaborado pelos autores 2014.

Na Figura 3.11 são mostrados os detalhes da dívida de uma determinada ação.

Destaque para a utilização da tag *center* para centralizar os botões Adicionar Despesas e Ver Dados do Parcelamento e para o destaque do números do RG e do CPF porque está instalado na máquina o programa SKYPE.

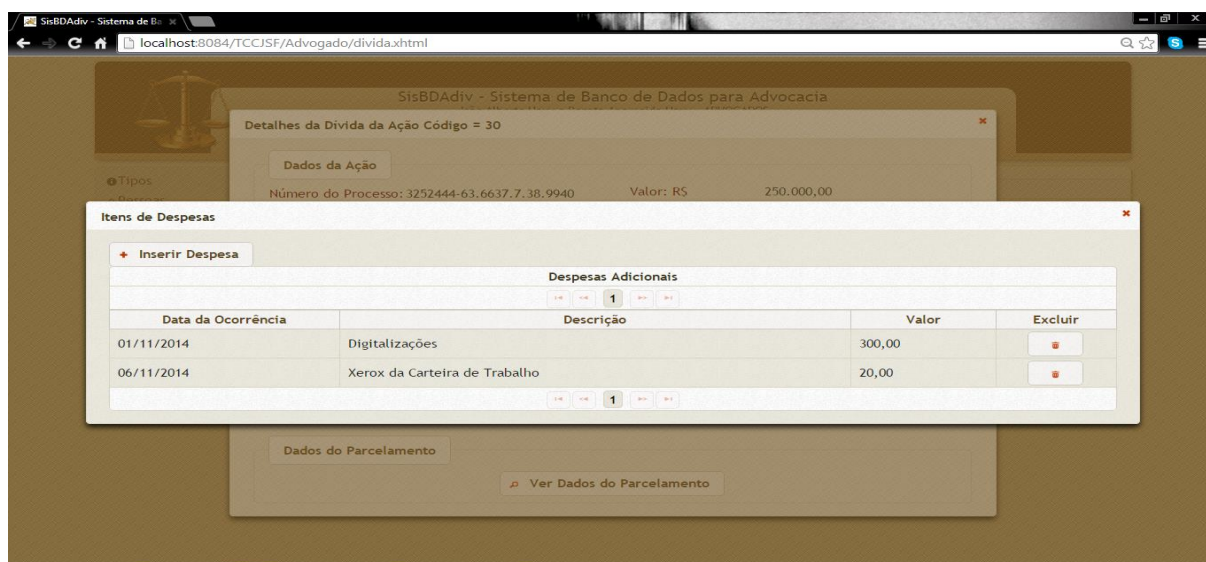


Figura 3.12 - Tela de Inserir Despesas ao Cliente
Fonte: Elaborado pelos autores 2014.

Na Figura 3.12 está destacada a tabela que lista as despesas adicionais do processo, estas o advogado deve receber juntamente com seus honorários.

Destaque para a utilização do atributo update no botão de inserir despesas que atualiza também os detalhes da dívida na tela que está atrás da tela da tabela.

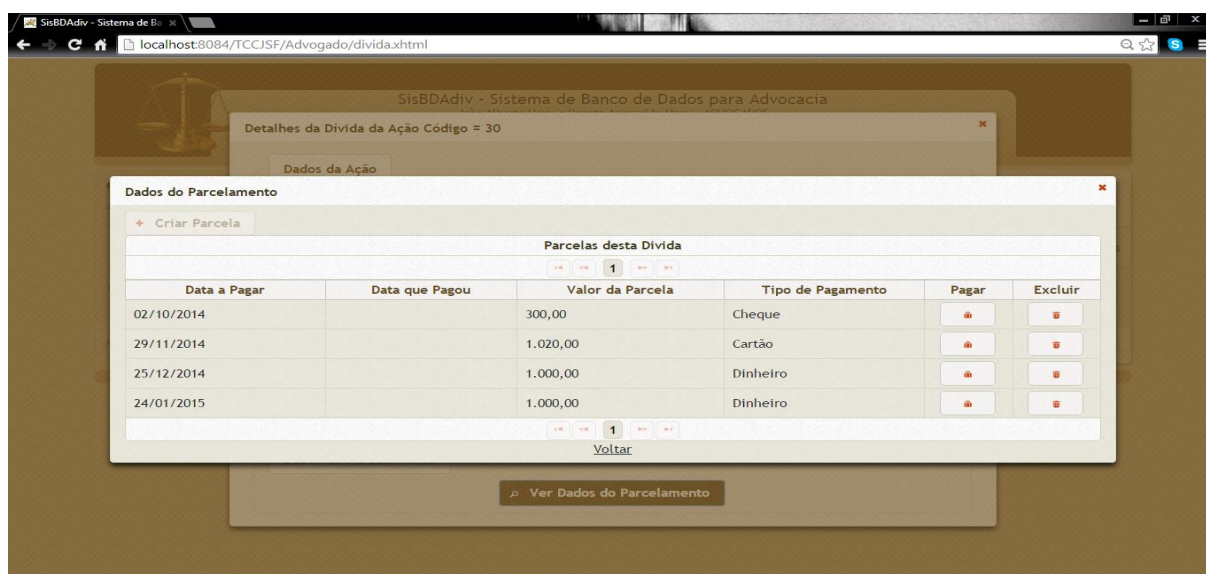


Figura 3.13 - Tela dos Dados de Parcelamento do Cliente
Fonte: Elaborado pelos autores 2014.

Na Figura 3.13 é mostrada a tela de parcelas que podem ser criadas ou pagas.

Destaque para o botão Criar Parcelas que está desabilitado por um método em Java que o desabilita caso o valor total da dívida tenha sido atingido.



		Relatório de Clientes Inadimplentes Escritório João Alberto Haury João Alberto Haury & Renata Aparecida Haury Advogados	
João		Ação:	41
	Valor da Parcela	Data a Pagar	
	R\$ 130,86	01/10/2014	
Luiza		Ação:	62
	Valor da Parcela	Data a Pagar	
	R\$ 3.000,00	30/10/2014	
	R\$ 30.000,00	01/10/2014	
	R\$ 40.000,00	01/10/2014	
Quinta-feira 30 Outubro 2014		Página 1 de 1	

Figura 3.14 - Tela Relatório de Clientes Inadimplentes
 Fonte: Elaborado pelos autores 2014.

Na Figura 3.14 tem-se o relatório criado em formato .pdf de clientes que possuem parcelas que não foram pagas até a data atual mas que deveriam ter sido pagas. Destaque para a utilização da IDE *IReport Designer* para elaboração do *design* do relatório e sua compilação para se tornar um arquivo .jasper antes de ser passado para .pdf pelo *servlet*.



Relatório de Faturamento Mensal

Escritório João Alberto Haury
João Alberto Haury & Renata Aparecida Haury Advogados

ANO 2014			
Mês de Janeiro			
Receitas		Despesas	
Código da Parcela	184	Código do Item	201
Data a Pagar	08/08/2014	Data da Despesa	02/01/14 00:00
Data que Pagou	02/01/2014	Valor	R\$ 30,00
Valor da Parcela	R\$ 2.631,50		
Código da Parcela	54		
Data a Pagar	31/12/2014		
Data que Pagou	17/01/2014		
Valor da Parcela	R\$ 3.000,00		
Código da Parcela	52		
Data a Pagar	18/10/2014		
Data que Pagou	18/01/2014		
Valor da Parcela	R\$ 3.400,00		
Código da Parcela	56		
Data a Pagar	30/01/2015		
Data que Pagou	29/01/2014		
Valor da Parcela	R\$ 200,00		
Total de Receitas	R\$ 9.231,50	Total de Despesas	R\$ 30,00
Ganho/Perda		R\$ 9.201,50	

Quinta-feira 30 Outubro 2014
Página 1 de 1

Figura 3.15 - Tela Relatório de Faturamento Mensal
Fonte: Elaborado pelos autores 2014.

Na Figura 3.15 está exposto o relatório de faturamento mensal que foi o relatório mais complexo de ser gerado visto que foi utilizada uma conexão JDBC com o banco dentro do servlet gerador e foram realizados diversos cálculos, dentre eles o de ano bissexto e o do valor do ganho/perda.

CONCLUSÕES

A área da advocacia atualmente está muito diferente do que podia-se ver há vinte anos atrás quando utilizava-se máquinas de escrever e grandes livros, durante esse tempo houve uma mudança nas maneiras de trabalhar com a ajuda da evolução tecnológica. No lugar da máquina veio o computador que além de um editor de texto trouxe mais tarde, com a internet uma proximidade com o mundo exterior.

Com isso houve também uma forma de sustentabilidade já que muitos papéis deixaram de ser utilizados, e foi gerada além de tudo uma facilitação nos serviços que antes eram feitos manualmente ou em máquinas de escrever e que hoje são feitos usando microcomputadores.

Além disso, os métodos de pesquisa em livros foram substituídos pela pesquisa online e ficou muito mais fácil encontrar aquilo que se deseja pelos avançados métodos de busca dos sites na internet.

Da mesma forma evoluiu também o jeito de organizar um escritório de advocacia, tanto as pastas e documentos quanto o andamento processual, hoje documentos são digitalizados e as pastas virtuais e o andamento de processos acompanhados por sistemas nos sites dos tribunais.

Portanto, com a criação deste novo sistema procurou-se atender as necessidades que foram encontradas na etapa de análise de requisitos, feita junto com os advogados do escritório.

Com relação às tecnologias, o serviço do banco de dados Oracle XE atendeu as necessidades de armazenamento dos dados bem como o gerenciamento de integridade dos mesmos. O *JDBC Oracle Driver* facilitou na conexão com a base de dados para a produção dos relatórios que foram necessários gerar a partir do sistema e também como conector para o Hibernate, visto que isto se dá por uma forma abstrata, ou seja, o mapeamento objeto-relacional é de certa forma escondido do usuário (programador), não são mostrados os detalhes em tempo de execução.

Utilizando o *framework* Java Server Faces juntamente com o Primefaces pôde-se encurtar o espaço de tempo gasto com componentes que já estão prontos e basta serem adaptados às funcionalidades para realizar a implementação da camada de visão do sistema.

Com o desenvolvimento do sistema o escritório passou a ter uma melhor organização de suas finanças que antigamente eram controladas por varias planilhas feitas no Microsoft Excel, o que se fazia necessário consultar para saber qual planilha se referia ao mês desejado, sendo que por meio do sistema de informação basta informar o mês e o ano que é gerado um relatório com as finanças mensais. Há também a possibilidade de analisar, por meio de outro relatório, os clientes que estão em dívida (inadimplentes). Os processos passaram a ter um acompanhamento dentro do sistema de informação, agilizando o acesso às informações necessárias para tomada de decisão para próximas alterações no processo (juntar alguns documentos, entrar com recursos, etc). Agora para cada cliente é gerado um relatório com as informações sobre todas as ações deste cliente contidas no escritório. Com relação ao acesso remoto o sistema notifica o outro usuário quando um usuário faz alterações (executa transações) ou faz login no sistema para sincronizar as informações e no final ter tudo atualizado.

Como trabalhos futuros propõe-se a preparação para o processo digital, e a integração com sites dos tribunais para que seja possível o protocolo de processos, a implementação do módulo secretário(a) que tornará possível a marcação de horários de atendimento de clientes juntamente com a de uma agenda para compromissos e também a do Gerenciador Eletônico de Documentos (GED) para tornar possível a organização de documentos sem a retenção dos mesmos no escritório causando mais acúmulo de papel.

REFERÊNCIAS BIBLIOGRÁFICAS

ABBEY, M.; ABRAMSOM, I.; COREY, M. **Oracle 8i Guia Introductório**. Rio de Janeiro: Campus, 2000.

BASTOS, D. F. **O que é Model View Controller (MVC)?**. 2011. Disponível em: <http://www.oficinadanet.com.br/artigo/desenvolvimento/o_que_e_model-view-controller_mvc> Acesso em: 17 nov. 2013.

BASHAM, B.; SIERRA, K.; BATES, B. **Use a Cabeça Jsp & Servlets**. 2. ed. Rio de Janeiro: Altabooks, 2008.

BEZERRA, A.; GOMES, M. **Programação Orientada a Objetos**. 2013. Disponível em: <http://www.metrooledigital.ufrn.br/aulas/disciplinas/poo/aula_15.html> Acesso em: 17 nov. 2013.

BRILL, G. **J2EE EJB, JDBC, JSP and Servlets**. Nova Yorke: Random House. 2013. Disponível em: <http://www.ceng.metu.edu.tr/~e1195288/CodeNotes_for_J2EE.pdf> Acesso em: 22 set. 2013.

COSTA E. Oracle em Versão Light. **Coleção Info Banco de Dados**. São Paulo, 27. ed. p. 36 – 37. 2006. Disponível em: <<http://downfacil.blogspot.com.br/2007/06/coleo-info-2006-banco-de-dados.html>> Acesso em 12 set. 2013.

DAVID, M. **Designing Rich Internet Applications**. Oxford: Elsevier, 2010.

DEITEL, P.; DEITEL, H. **Java Como Programar**. Tradução Edson Furmankievicz. 8. ed. São Paulo: Pearson, 2010.

FILHO, M. P. C. Revista de Administração de Empresas. **Os sistemas de informação e as modernas tendências de tecnologia e dos negócios**. São Paulo, v. 34, n. 6, p. 33-45, Nov/Dez. 1994. Disponível em: <http://rae.fgv.br/sites/rae.fgv.br/files/artigos/10.1590_S0034-75901994000600005.pdf> Acesso em: 10 ago. 2013.

GOMES, Y. M. P. **Java na Web com JSF, Spring, Hibernate e Netbeans 6**. Rio de Janeiro: Editora Ciência Moderna, 2008.

GONÇALVES, E. **Dominando NetBeans**. 1. ed. Rio de Janeiro: Editora Ciência Moderna, 2006.

GRANJEIA, M. A. L. **Desenvolvimento de uma aplicação web para pequenos escritórios de advocacia, combinado com uso de dispositivo móvel**. 2011. TCC (Graduação em Banco de Dados) – FATEC, Lins/SP.

JAVA. **O que é a Tecnologia e por que é Necessária**. Disponível em: <https://www.java.com/pt_BR/download/faq/whatis_java.xml> Acesso em: 20 out. 2013.

JUNEAU, J. Primefaces na Empresa. Disponível em: <<http://www.oracle.com/technetwork/articles/java/java-primefaces-2191907.html>> Acesso em: 10 jan 2014.

K19. **Desenvolvimento Web com JSF2 e JPA2**. 2013. Disponível em: <<http://pt.scribd.com/doc/55978445/k19-k12-desenvolvimento-web-com-jsf2-e-jpa2>> Acesso em: 20 mar. 2014.

LEE, R. W. **Pesquisa jurisprudencial inteligente**. 1998. Tese (Doutorado em Engenharia) – Programa de Pós-Graduação em Engenharia de Produção, Universidade de Santa Catarina, Florianópolis.

MACEDO, A. Relatórios em Java – JasperReports e iReport. 2010. Disponível em: <<http://www.k19.com.br/artigos/relatorios-em-java-jasperreports-e-irepor/>> Acesso em: 25 mar 2014.

MACHADO, F. N. R. **Banco de dados projeto e implementação**. 2. ed. São Paulo: Érica Ltda., 2010.

NASSU, E. A.; SETZER, V. W. Bancos de dados orientados a objetos. 2. ed. São Paulo: Edgard Blücher, 1999.

NETBEANS. **NetBeans IDE a Forma Mais Inteligente e Rápida de Codificar**. 2013. Disponível em: <https://netbeans.org/features/index_pt_BR.html> Acesso em: 15 out. 2013.

NIEDERAUER, JULIANO. **Web Interativa com Ajax e PHP**. 2. ed. São Paulo: Novatec, 2013.

OLIVEIRA, D. P. R. **Sistemas organização & métodos: uma abordagem gerencial**. 3. ed. São Paulo: Atlas, 1990.

ORTONCELLI, A. R.; LAZARIN C. A. J. **Persistência de dados com Java e Hibernate Annotations**. 2009. 37p. Universidade Estadual do Norte do Paraná. Disponível em: <www.atualizado.com.br/semfronteiras/index.php?option=com_docman&task=doc_download&gid=8&itemid=66>

RICARTE, I. L. M. **Programação Orientada a Objetos: Uma abordagem com Java**. 2001. Disponível em: <<http://www.dca.fee.unicamp.br/courses/PooJava/Aulas/poojava.pdf>> Acesso em: 17 nov. 2013

SIERRA, K.; BATES, B. **Use a Cabeça! Java**. 2. ed. Rio de Janeiro: Altabooks, 2005.

SILVA, A. C. **Unified Modeling Language (UML)**. 2013. Disponível em: <http://www.deinf.ufma.br/~acmo/MOO_Classe.pdf> Acesso em: 18 nov. 2013.

SILVA, M. S. **CSS3 Desenvolva Aplicações Web Profissionais Com Uso Dos Poderosos Recursos De Estilização Das Css3**. 2. Ed. São Paulo: Novatec, 2012.

SILVA, M. S. **JQuery a Biblioteca do Programador JavaScript**. 2. Ed. São Paulo: Novatec, 2013.

SILVA, M. S. **JavaScript: Guia do Programador**. 2. ed. São Paulo: Novatec, 2010.

STAIR, R. M. **Princípios de sistemas de informação**. 2. ed. Rio de Janeiro: LTC, 1998.

TEOREY, T.; LIGHTSTONE, S.; NADEAU, T. **Projeto e modelagem de bancos de dados**. Tradução Daniel Vieira. 4. ed. Rio de Janeiro: Elsevier, 2007.