



Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis
Campus "José Santilli Sobrinho"

JOSÉ AMÉRICO DOICHE JUNIOR

INFORMATIZAÇÃO DOS SERVIÇOS NOTARIAIS

**Assis/SP
2012.**

INFORMATIZAÇÃO DOS SERVIÇOS NOTARIAIS

Trabalho de Conclusão de Curso apresentado ao curso de Bacharelado em Ciência da Computação do Instituto Municipal de Ensino Superior de Assis – IMESA e a Fundação Educacional do Município de Assis – FEMA, como requisito parcial à obtenção do Certificado de Conclusão de Curso.

Orientando: José Américo Doiche Junior

Orientador: Prof. Dr. Luiz Ricardo Begosso

FICHA CATALOGRÁFICA

DOICHE JUNIOR, José Américo.
Informatização dos Serviços Notariais/ José Américo Doiche Junior.
Fundação Educacional do Município de Assis - FEMA -- Assis, 2012.
46 f.

Orientador: Prof. Dr. Luiz Ricardo Begosso.
Trabalho de Conclusão de Curso - Instituto Educacional do Ensino Superior de Assis - IMESA

1. Cartórios 2. Informatização 3. Serviços.

CDD: 001.61
Biblioteca da FEMA

INFORMATIZAÇÃO DOS SERVIÇOS NOTARIAIS

JOSÉ AMÉRICO DOICHE JUNIOR

Trabalho de Conclusão de Curso
apresentado ao Instituto de Ensino Superior
de Assis, como requisito do Curso de
Bacharelado em Ciência da Computação.

Orientador: Prof. Dr. Luiz Ricardo Begosso.

Analizador: Felipe Alexandre Cardoso Pazinatto

Assis
2012.

DEDICATÓRIA

Dedico este trabalho à minha esposa Aline, que muito me ajudou durante a minha formação acadêmica, aos meus filhos Joaquim e David José que me dão motivos para seguir em frente.

AGRADECIMENTOS

Agradeço a Deus que sempre esteve comigo, me ajudando, amparando e me dando forças, principalmente quando me senti completamente sem forças e sem esperanças.

Agradeço à minha esposa Aline, que me apoiou muitas vezes, e que acreditou em meu potencial, mesmo quando eu não acreditava mais. Agradeço ao meu filho Joaquim que com seus sorrisos me fortalecia dia após dia, renovando o meu ânimo. Agradeço ao meu filho David José que renovou minhas forças para seguir em frente.

Agradeço aos meus pais José Américo e Ivone, que criaram as bases da minha personalidade, e sem os quais eu não seria quem sou.

Agradeço aos meus tios Lourival e Sandra, que tornaram realidade o meu sonho de fazer faculdade, e que muito me ajudaram e apoiaram.

Agradeço aos meus sogros Claudionor e Ana Maria, que me deram muitas palavras de estímulo e incentivo, me ajudando a sempre prosseguir.

Agradeço ao meu professor e orientador Luiz Ricardo Begosso, que teve a paciência e a disponibilidade de me orientar principalmente quando eu estava perdido.

Agradeço a todos os professores que transferiram seus conhecimentos a mim, nos quais certamente me espelharei, em especial ao meu professor Fernando Lima que me ajudou quando mais precisei.

Agradeço todos os meus amigos de faculdade, em especial o Wellington que foi meu companheiro de estudos.

“Se hoje fosse o último dia de minha vida, queria fazer o que vou fazer hoje? E se a resposta fosse ‘não’ muitos dias seguidos, sabia que precisava mudar algo”.

Steve Jobs

RESUMO

Este trabalho tem por objetivo demonstrar como é feita a informatização dos serviços de um cartório, a partir da informatização de apenas um dos diversos setores existentes na serventia. Priorizou-se informatizar a parte de notas, que diz respeito ao reconhecimento de firmas, um dos serviços mais procurados nos cartórios em geral. Conceitos teóricos são apresentados de forma resumida e simplificada para que todos os leitores, familiarizados ou não com esses conceitos possam compreender sua importância e aplicabilidade no presente trabalho. Dessa forma, buscou-se unir o desenvolvimento de um software que informatizasse um dos serviços mais buscados nos cartórios.

Palavras-chave: Cartórios, Informatização, Serviços.

ABSTRACT

This paper aims to show how is the computerization of the services of a notary, from the computerization of only one of several existing sectors in usefulness. Prioritized to computerize some of the notes, as regards the recognition of firms, one of the most sought after in notary services in general. Theoretical concepts are presented in summary form and simplified so that all readers are familiar with these concepts or not they understand its importance and applicability in the present work. Thus, we sought to unite the development of software that computerize one of the most commonly sought in the registries.

Keywords: Registries, Information Technology, Services.

RESUMEN

Este trabajo pretende mostrar cómo es la informatización de los servicios de un notario, a partir de la informatización de sólo uno de los sectores existentes en la utilidad. Prioridad a la informatización de algunas de las notas, lo que se refiere al reconocimiento de las empresas, uno de los más buscados después de los servicios notariales en general. Los conceptos teóricos se presentan en forma resumida y simplificada para que todos los lectores estén familiarizados con estos conceptos o no entienden su importancia y aplicabilidad en el presente trabajo. Por lo tanto, hemos tratado de unir el desarrollo de software que informatizase uno de los que más se utilizan en los registros.

Palabras clave: Registros, Tecnología de la Información, Servicios.

ÍNDICE DE DIAGRAMAS

Diagrama 1 – Diagrama de casos de uso.....	32
Diagrama 2 – Diagrama de classes.....	36
Diagrama 3 – Diagrama de entidade e relacionamentos.....	38
Diagrama 4 – Diagrama de Atividades.....	39

ÍNDICE DE FIGURAS

Figura 1	– Representação gráfica de um ator do diagrama de casos de uso.....	30
Figura 2	– Representação gráfica do relacionamento entre ator e caso de uso..	31
Figura 3	– Caso de uso: Manter funcionários.....	33
Figura 4	– Caso de uso: Manter clientes.....	33
Figura 5	– Caso de uso: Manter reconhecimentos de firma.....	34
Figura 6	– Caso de uso: Relatório de reconhecimento de firmas.....	34
Figura 7	– Caso de uso: Relatório de clientes.....	35
Figura 8	– Caso de uso: Relatório de funcionários.....	35
Figura 9	– Print da tela do software DBDesigner.....	37
Figura 10	– Tela inicial do sistema.....	40
Figura 11	– Cadastro de funcionários.....	41
Figura 12	– Cadastro de clientes.....	41
Figura 13	– Reconhecimento de assinaturas.....	42
Figura 14	– Relatório de reconhecimento de firmas.....	42
Figura 15	– Relatório de clientes cadastrados.....	43
Figura 16	– Relatório de funcionários.....	43

SUMÁRIO

1. INTRODUÇÃO	15
2. JUSTIFICATIVAS DO SISTEMA	17
2.1 - A tendência da informatização e a inserção do projeto	17
2.2 - Dúvidas com relação à viabilidade do projeto.....	17
2.3 - Objetivos a serem alcançados	17
2.4 - Considerações gerais	18
3. CONCEITOS TEÓRICOS	19
3.1 - Definindo sistema	19
3.2 - Casos de uso e requisitos.....	19
3.2.1 - Especificações de requisitos.....	20
3.3 - Garantia de qualidade do software	21
3.4 - UML	22
3.5 - Casos de uso	23
3.6 - Classes	24
4. MÉTODOS DE DESENVOLVIMENTO	25
4.1 – Ferramentas.....	25
4.2 - Java	25
4.3 - Eclipse IDE	26
4.4 - ArgoUML	26
4.5 - MySQL.....	26
4.6 - DBDesiner.....	27
5. ANÁLISE DO SISTEMA	28
5.1 – Dados sobre o cliente	28
5.2 – Coleta de requisitos	28
6. DIAGRAMAS	30
6.1 – Diagrama de casos de uso – definição	30

6.1.1 – Diagrama de casos de uso – aplicação.....	32
6.2 - Descrição de cada um dos casos de uso do sistema	32
6.2.1 – Manter funcionários.....	33
6.2.2 – Manter clientes	33
6.2.3 – Movimentar reconhecimento de firmas	33
6.2.4 – Relatório de reconhecimento de firmas.....	34
6.2.5 – Relatório de clientes.....	34
6.2.6 – Relatório de funcionários.....	35
6.3 – Diagrama de classes - definição	35
6.3.1 – Diagrama de classes – aplicação.....	36
6.4 – Diagrama de entidade e relacionamento - definição	37
6.4.1 – Diagrama de entidade e relacionamento – aplicação.....	38
6.5 – Diagrama de atividades/ - definição	38
6.5.1 – Diagrama de atividades – aplicação.....	39
7. TELAS DO SISTEMA	40
7.1 – Tela inicial do sistema	40
7.2 – Cadastro de funcionários	41
7.1 – Cadastro de clientes	41
7.1 – Reconhecimento de Assinaturas.....	42
7.1 – Relatório de reconhecimento de firma	42
7.1 – Relatório de clientes cadastrados	43
7.1 – Relatórios de funcionários.....	43
CONSIDERAÇÕES FINAIS	44
REFERÊNCIAS.....	46

1. INTRODUÇÃO

Atualmente é extremamente necessário informatizar os processos de quaisquer tipos de estabelecimentos prestadores de serviços. A valorização do tempo tem ajudado para que todos os setores sofram informatização em seus serviços, uma vez que a partir da informatização os processos tornam-se menos demorados.

A correria do dia-a-dia fornece aos profissionais de T.I. (Tecnologia da Informação) possibilidades de facilitar o cotidiano das pessoas, criando e disponibilizando tecnologias cada vez mais voltadas a essas necessidades.

Um sistema informatizado permite que cadastros sejam preenchidos com mais rapidez e precisão, além de facilitar a consulta de dados, entre outras funcionalidades agilizando uma parte dos serviços do cartório.

Será criado um sistema comercial voltado para as necessidades do ^{1º} *Tabelião de Notas e Protestos, Títulos e Letras de Assis*, cujo atual Tabelião é o senhor Lourival Gama da Silva. Atualmente o cartório conta com 15 funcionários, sendo que 7 são responsáveis pelas escrituras, procurações, inventários, entre outros e 8 são responsáveis pelo atendimento geral, que realiza serviços de autenticações, reconhecimentos de firma, cópias, protestos, entre outros.

No item 1 são descritas as justificativas do sistema, ou seja, porque é necessário criar um sistema para o cartório em questão, onde são dissertados temas como a tendência da informatização dos serviços, a inserção do projeto proposto, As dúvidas relativas à viabilidade do projeto, os objetivos e o público ao qual o sistema é destinado, as motivações do projeto e as ferramentas utilizadas para a fomentação do projeto e do sistema.

Os conceitos teóricos são apresentados no item 3, cujo objetivo é descrever esses conceitos de maneira clara, concisa e breve, de modo que até mesmo os leitores que não estão inseridos no ramo de ciência da computação possam compreendê-los.

O item 4 é onde são discutidos os métodos de desenvolvimento do sistema, enfocando as ferramentas utilizadas tanto para a criação do software quanto do trabalho escrito. As ferramentas Java, Eclipse IDE, ArgoUML, MySQL e DBDesigner Fork são brevemente descritas e seu uso é justificado.

O item 5 é uma breve análise do cliente do sistema, cujos dados foram coletados, juntamente com os requisitos desejados pelo cliente.

No item 6 são apresentados os diagramas de Casos de Uso, de Classes, de Entidade e Relacionamento e de Atividades, onde é possível ao leitor visualizar como o sistema deverá funcionar após a sua conclusão.

Em seguida, é apresentado o item 7, onde estão colocadas imagens das telas do sistema. Após esses capítulos, uma breve conclusão é apresentada, relatando como foi a criação do sistema.

2. JUSTIFICATIVAS DO SISTEMA

2.1 A Tendência da Informatização e Inserção do Projeto

A tendência atual é que todos os setores sofram informatização de seus serviços. Já existem cartórios que fazem uso desse tipo de atendimento, o que denota ser possível criar e implantar tal projeto. Para tanto, é necessário que haja interesse por parte do dono do cartório em utilizar tal sistema. Inicialmente o sistema irá oferecer poucas possibilidades de serviços, mas com o tempo poderá ser ampliado e melhorado de acordo com as necessidades do cartório.

Já a qualidade dos serviços prestados dependerá de um bom levantamento de requisitos e de uma análise e interpretação detalhada desse levantamento. A agilidade do sistema será decorrente de sua programação, que por sua vez depende da correta utilização dos requisitos coletados e analisados.

2.2 Dúvidas com Relação à Viabilidade do Projeto

As perguntas mais latentes do projeto são as seguintes:

- É possível criar um software para o cartório de maneira que atenda às necessidades do tabelião e seus funcionários?
- Este software possuirá condições de oferecer qualidade e agilidade no atendimento?

Essas perguntas serão respondidas à medida que o Sistema for sendo desenvolvido, e serão apresentadas na conclusão do projeto.

2.3 Objetivos a Serem Alcançados

O objetivo desse trabalho é desenvolver um pequeno sistema para 1º Cartório de Notas e Protestos de Assis, de modo que facilite e agilize os processos de reconhecimento de firmas, um dos serviços mais procurados nesse estabelecimento.

Além desse objetivo geral, existem os objetivos específicos que são: desenvolver um sistema com ênfase no reconhecimento de firmas, tornado possível a visualização da imagem das fichas de assinatura

2.4 Considerações Gerais

O tabelião e seus funcionários serão o público alvo deste sistema, que serão beneficiados diretamente pela agilidade na prestação de serviços. Os clientes do cartório também serão beneficiados, mesmo que indiretamente, por esse software, que tornará o atendimento mais rápido e prático. O que se espera é um sistema que armazene as informações e que permita buscas rápidas.

3. CONCEITOS TEÓRICOS

3.1 Definindo Sistema

Para de definir o sistema é necessário que sejam encontrados os atores, fazendo perguntas como:

- Quem irá usar o sistema?
- Quem irá instalar o sistema?
- Quem irá fazer manutenção dos dados desse sistema?
- Quem irá desligar o sistema?

Entre diversas outras perguntas. Além de encontrar os atores, é necessário encontrar os casos de uso, através de perguntas como:

- Quais funções foram requeridas pelos atores?

Um glossário com os termos que descrevem as funcionalidades e as características do sistema também deve ser criado, e os requisitos não funcionais devem ser coletados. Martins (2007, p. 213) explica a definição de um sistema:

Com base nas informações providas pelos stakeholders, documentar os recursos e as características do sistema que será desenvolvido. Definir o critério que será seguido para priorizar os requisitos a serem implementados e administrar as expectativas dos stakeholders quanto à possibilidade de implementação dos requisitos. As funcionalidades chaves devem ser documentadas através de casos de uso.

3.2 Casos de Uso e Requisitos

Casos de uso representam a interação entre um usuário (que pode ser uma pessoa ou uma máquina) e o sistema. Cada um dos casos de uso irá descrever uma funcionalidade do sistema. Segundo Ivan Jacobson, caso de uso “*é um documento narrativo que descreve 3 seqüência de eventos de um ator que usa um sistema para completar um processo*”.

Isso quer dizer que os casos de uso são um tipo de técnica de modelagem utilizada para descrever o que um novo sistema é capaz de fazer. Para se formar os casos de uso é necessário que haja um processo interativo entre o cliente e o

desenvolvedor do sistema, de forma que haja discussões que levem às especificações que o cliente deseja e que o desenvolvedor seja capaz de criar para o sistema em questão. Martins (2007, p.206) fala sobre a documentação dos casos de uso:

[...] Os processos são documentados através de casos de uso, as pessoas e sistemas através de atores. Os diagramas de atividades podem ser utilizados para descrever os casos de uso junto com as descrições funcionais. As necessidades do negócio é que vão definir a especificação e os requisitos do novo sistema. (Martins, 2007, p.206)

Os casos de uso facilitam a formalização das funções que serão realizadas pelo sistema, e é caracterizado por ser uma lista completa (do início ao final da tarefa) de todas as interações possíveis entre o usuário e o sistema, de maneira que uma tarefa seja cumprida.

É importante que os casos de uso sejam compreensíveis aos usuários, uma vez que eles são os responsáveis por determinar o que o sistema deve ou não fazer. Os casos de uso permitem visualizar se o usuário e o desenvolvedor estão de acordo com os recursos que o sistema deve disponibilizar.

3.2.1 Especificações de requisitos

As especificações de requisitos são necessárias para se obter sistemas de melhor qualidade, e que satisfaçam às necessidades dos clientes tanto em funções quanto em prazo e orçamento.

O requisito pode ser considerado uma função, uma propriedade ou uma restrição que deve ser incluída no sistema durante a sua modelagem. Isso quer dizer que os requisitos são uma descrição de um serviço ou de uma limitação do sistema. Paula Filho (2009, p.8) fala sobre os requisitos:

Mesmo requisitos documentados podem apresentar problemas. Uma especificação de requisitos pode conter requisitos incompletos, inconsistentes ou ambíguos. Alguns desses problemas decorrem da própria linguagem natural, que normalmente é usada para expressá-los. Outros decorrem de técnicas deficientes de levantamento e especificação dos requisitos. (Paula Filho, 2009, p. 8)

Uma das principais causas de fracasso de um projeto de software são as especificações de requisitos mal formuladas, o que gera constantes alterações dos requisitos. Isso quer dizer que durante as etapas iniciais do desenvolvimento as probabilidades de surgirem problemas são maiores devido à má formulação dos requisitos, que deve ser feita de maneira clara e voltada às necessidades dos usuários do sistema.

Para que a especificação de requisitos seja bem formulada é preciso que o programador faça um levantamento prévio de tudo o que o cliente espera do software, e trabalhe os requisitos de acordo com essas necessidades.

3.3 Garantia de Qualidade do Software

A garantia da qualidade do software é obtida através da normatização dos processos de desenvolvimento do sistema. Os modelos utilizados para se garantir a qualidade dos softwares atuam no processo, mas o principal objetivo da garantia da qualidade do software é garantir que o software final esteja coerente com as expectativas do cliente, ou seja, de acordo com o que foi definido a partir da coleta e análise dos requisitos. Paula Filho (2009, p.11) fala a respeito da garantia de qualidade do software:

Em todas as fases do desenvolvimento de software, as pessoas introduzem defeitos. Eles decorrem de limitações humanas: erros lógicos, erros de interpretação, desconhecimento de técnicas, falta de atenção ou falta de motivação. Em todo bom processo, existem atividades de garantia da qualidade, tais como revisões, testes e auditorias. Essas atividades removem parte dos defeitos introduzidos. Quando atividades de corte de garantia da qualidade são cortadas, parte dos defeitos deixa de ser removida em um ponto do projeto. (Paula Filho, 2009, p. 11)

Durante o desenvolvimento do software a sua qualidade está relacionada à qualidade do processo de desenvolvimento, sendo assim, torna-se comum buscar softwares de maior qualidade através dos processos de desenvolvimento de softwares.

3.4 UML

UML significa *Unified Modeling Language*. É uma linguagem de modelagem não proprietária que auxilia na visualização dos relacionamentos entre os objetos. Ou seja, permite que o desenvolvedor de um software visualize seus trabalhos em diagramas padronizados.

A UML não é um método de desenvolvimento, por isso não fornece o passo a passo de como projetar um sistema. Ela surgiu da necessidade de padronização da análise e da modelagem dos sistemas orientados a objetos.

Em 1997, seus três desenvolvedores: Grady Booch, James Raumbaugh e Ivar Jacobson se uniram com o intuito de unificar suas metodologias, selecionando o melhor de cada uma delas para poderem obter a UML. Atualmente é mantida pela OMG – *Object Management Group*.

Por se tratar de uma modelagem de sistemas, pode ser usada para especificar, modelar, visualizar e documentar todas as fases do desenvolvimento do software (análise de requisitos, análise de sistema, design, programação e testes).

A análise dos requisitos deve coletar as necessidades dos usuários do sistema através das funções contidas nos casos de uso. A análise é a fase onde são criadas as abstrações e mecanismos através da interpretação dos requisitos.

O design nada mais é do que o resultado obtido pela análise, ou seja, é o detalhamento dos problemas, necessário para a fase de programação. Durante a programação os modelos e problemas são descritos em linguagem de programação. Os testes servem para verificar a funcionalidade da programação. Podem ser unitários, de integração e de aceitação.

É de extrema importância desenvolver o modelo de um software, pois facilita a comunicação entre o desenvolvedor e o cliente/usuário, além de proporcionar uma arquitetura visual mais bonita.

As divisões da UML são:

- Visões: os diferentes aspectos do sistema são mostrados através de níveis de abstrações diferentes, de modo que seja criada uma visão completa do sistema que será construído.

- Modelos de elementos: são os conceitos que norteiam a criação dos diagramas.
- Mecanismos gerais: semântica ou informações a respeito dos elementos do projeto.
- Diagramas: são uma visão gráfica dos conteúdos do projeto.

Usar a UML torna mais fácil o processo de criação e desenvolvimento do um projeto, uma vez que facilita a visualização de todas as etapas que o projeto deverá passar.

3.5 Casos de Uso:

São diagramas de desenvolvimento de softwares que determinam as especificações funcionais de um sistema. Paula Filho (2009, p. 178-9) fala sobre os casos de uso:

Os casos de uso representam funções completas do produto. Um caso de uso realiza um aspecto maior da funcionalidade do produto: todo caso de uso deve gerar um ou mais benefícios para o cliente ou os usuários, visíveis nas pós-condições resultantes. O conjunto de casos de uso deve descrever a funcionalidade completa do produto, sem lacunas e sem superposições; daí dizer-se que cada caso de uso representa uma “fatia de funcionalidade”. (Paula Filho, 2009, p; 178-9).

Os diagramas auxiliam a comunicação entre os desenvolvedores do software e seus clientes através da descrição de um cenário que enumera as funcionalidades do sistema a partir do ponto de vista do usuário, sendo que são apresentadas ao cliente as principais funcionalidades do sistema.

Os diagramas de casos de uso são representados por atores, casos de uso, relacionamentos entre os elementos. Os atores são usuários do sistema, que podem ser humanos ou até mesmo um sistema operacional. O caso de uso descreve uma função do sistema. Já os relacionamentos facilitam a descrição dos casos de uso.

Os clientes utilizam os casos de uso para compreender como o sistema se comporta, aprovando ou não os resultados obtidos através da sua modelagem. Já os arquitetos dos softwares utilizam os casos de uso para identificar as

funcionalidades da arquitetura e os desenvolvedores do software compreendem e refinam o comportamento do sistema através dos casos de uso.

Sua importância se deve ao fato de que todos os atores envolvidos no processo de concepção, criação e finalização de um software se beneficiam dos casos de uso, facilitando suas atividades.

3.6 Classes

Classes são definidas como um conjunto de objetos cujos comportamentos e características são comuns. Essas características comuns são voltadas para a programação, e é necessário que se definam suas propriedades e seus métodos. Isso quer dizer que uma classe é uma forma de abstrair objetos do mundo real transportando para o mundo digital, ou seja, uma forma de escrever os objetos

As propriedades são características dos objetos que estão sendo representados pelas classes. Já os métodos são as ações que os objetos podem realizar. Isso quer dizer que as classes irão especificar tanto a estrutura quanto as operações dos objetos. Os diagramas de classes são os principais diagramas da UML, já que apontam as classes, as interfaces e os relacionamentos entre essas classes, além de ilustrar os atributos e operações de uma determinada classe, descrevendo os tipos de objetos presentes no sistema bem como os relacionamentos entre eles.

4. MÉTODOS DE DESENVOLVIMENTO

4.1 Ferramentas

As ferramentas utilizadas para o desenvolvimento desse sistema serão: Java, Eclipse, ArgoUML, MySQL, DB Designer Fork 2009.

4.2 Java

A linguagem adotada para o desenvolvimento do software é a Java, que é uma linguagem de programação orientada a objetos e muito confiável. Com a linguagem Java é possível construir desde sistemas críticos até sistemas que necessitem de altíssima velocidade para funcionar.

Diversas empresas do ramo tecnológico participam e patrocinam a Java, por isso é considerada uma das melhores linguagens de programação devido a algumas características como portabilidade, robustez, segurança, orientação a objetos, alto desempenho, dinamismo e facilidade.

A Java é uma linguagem interpretada, e por isso, pode ser executada em diferentes plataformas ou equipamentos que possuam o interpretador Java compilado para o sistema operacional.

Na linguagem Java, tanto os seus recursos quanto o seu ambiente trabalham no tempo de execução, característica que garante ao usuário que o sistema não sofrerá uma queda.

Por se tratar de uma linguagem orientada a objetos, os recursos de herança e reutilização de códigos ocorrem tanto de forma dinâmica quanto estática. O alto desempenho da linguagem Java é obtido graças à possibilidade de suportar diversos recursos de alto desempenho. Já o dinamismo é decorrente do seu projeto ser orientado a objetos, permitindo a extensibilidade durante a execução.

Por ser uma linguagem que foi derivada da linguagem C e C++ ela torna-se familiar e o próprio ambiente se responsabiliza pelo gerenciamento da memória e dos ponteiros, eliminando essa responsabilidade do programador.

4.3 Eclipse IDE

A Eclipse foi escolhida pois é uma IDE (*Integrated Development Environment* ou Ambiente Integrado de Desenvolvimento) desenvolvida em Java, cujo código é aberto para a criação de softwares. A IBM foi quem desenvolveu a primeira versão Eclipse, e o doou como software livre.

Atualmente é a IDE mais utilizada no mundo, utiliza SWT como biblioteca gráfica no lugar de Swing. Através de seus plug-ins pode ser desenvolvido em Java, em C / C++, PHP, entre outras.

4.4 ArgoUML

A ArgoUML é uma aplicação que serve para modelar o desenho do software. Por ser implementada em Java roda na maior parte das plataformas. Foi escolhida como aplicação de modelagem do software para desenvolver os diagramas de casos de uso, os diagramas de classes e os de atividades.

Os diagramas de casos de uso demonstram, através de uma visão geral, as funcionalidades do sistema e o diagrama das classes serve para facilitar a análise do relacionamento entre os objetos, facilitando a visualização entre os objetos.

4.5 MySQL

O MySQL foi escolhido como banco de dados deste trabalho devido a algumas características que o diferem da maioria dos outros bancos de dados. Além de ser um dos bancos de dados mais populares, sendo usado por grandes empresas como a NASA, a Cisco Systems, a Nokia, entre outros, ele é facilmente integrado com o PHP que é incluído nos pacotes de sites de hospedagem da Internet.

As características diferenciais desse banco de dados são:

- Portabilidade (suporta quase todas as plataformas atuais);

- Compatibilidade com diversas linguagens de programação, entre elas: Delphi, Java, C/C++, ASP, Perl, Visual Basic, Ruby, etc.;
- Desempenho e estabilidade;
- Exige poucos recursos de *hardware*;
- É um *software* livre (GLP);
- Possui facilidade de uso;
- Suporta *Stored Procedures e Functions*, controle transacional, *Triggers, Cursors, etc.*
- As interfaces gráficas são fáceis de utilizar e foram cedidas pela MySQL Inc. (MySQL Toolkit)

Tornou-se interessante utilizar este bando de dados no desenvolvimento do sistema devido ao seu código ser aberto e por funcionar em sistemas operacionais como Linux, Windows, SunOS, entre outros. Além do mais, possui alto desempenho e robustez e é caracterizado por ser multitarefa e multiusuário. Por se tratar de um banco de dados relacional, isto é armazenar os dados em tabelas separadas ao invés de colocar todos os dados em um mesmo local, ele é considerado veloz e flexível.

3.6 DBDesigner

O DBDesigner é importante na modelagem do banco de dados, pois sua interface integra os diagramas, a modelagem, a construção e a manutenção de base dos dados. No presente trabalho tem importância na modelagem do diagrama de entidade e relacionamentos.

Para desenhar as tabelas, os relacionamentos e aos atributos foi criada uma interface simples, que além de ser descomplicada proporciona ao programador uma modelagem mais detalhada.

5. ANÁLISE DO SISTEMA

5.1 Dados sobre o cliente:

Empresa: Primeiro Tabelião de Notas de Protesto de Letras e de Títulos de Assis.

Tabelião responsável: Lourival Gama da Silva.

Endereço: Rua Floriano Peixoto 695 Centro Assis – SP.

Telefone: (018) 3302-1597

5.2 Coleta de requisitos

Através de entrevista realizada com o tabelião foram definidos os requisitos necessários para o desenvolvimento do sistema desktop:

- Cadastrar clientes
- Cadastrar funcionários
- Reconhecer firmas

A partir desta entrevista, o sistema foi dividido em três partes: Cadastros, Movimentações e Relatórios.

Cadastrar / Manter:

1. Funcionários
2. Clientes

Movimentar:

1. Reconhecimento de firmas

Relatórios:

1. Reconhecimento de firmas
2. Cliente

3. Funcionário

A seguir será apresentada uma lista simplificada de eventos que poderão ocorrer no sistema:

Lista de eventos	
Cadastrar / Manter	<ul style="list-style-type: none">• Funcionários• Clientes
Movimentar	<ul style="list-style-type: none">• Reconhecimento de firmas
Emitir relatórios de	<ul style="list-style-type: none">• Reconhecimento de firmas• Cliente• Funcionário

6. DIAGRAMAS:

A seguir serão apresentados três tipos de diagramas: diagrama de casos de uso, diagrama de classes e diagrama de entidade e relacionamento.

6.1 Diagrama de Casos de Uso – Definição

O diagrama de casos de uso (também conhecido como *use case*) representa uma unidade funcional do sistema, ou seja, é uma representação das sequências de eventos de um ator.

O objetivo de um diagrama de casos de uso é demonstrar de maneira simples e compreensível as funcionalidades de um sistema às pessoas que não possuem familiaridade com o universo da informática/programação.

Além disso, o diagrama de casos de uso auxilia a análise do sistema, especificando tanto o comportamento quanto as funcionalidades do sistema. São úteis na delimitação do sistema e servem de base para derivar casos de testes.

Os atores do diagrama de casos de uso são elementos externos ao sistema que de alguma forma participam desse diagrama, estimulando eventos no sistema, tanto de entrada como de saída.

Os atores são designados através do papel por eles exercido no caso de uso, podendo ser o programador, o cliente, o usuário, o operador, etc. São designados pelo desenho de um boneco, como o apresentado a seguir:

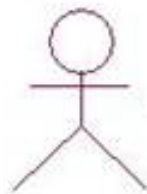


Figura 1. Representação gráfica de um ator do diagrama de casos de uso.

Para cada caso de uso, existe um ator que deve iniciá-lo, e esse ator deve ser indicado explicitamente, de maneira que sejam facilmente identificáveis.

Os atores são descritos da seguinte forma:

- Pessoas
- Softwares ou sistemas de computação
- Dispositivos elétricos e/ou mecânicos
- Hardware

Dentro do diagrama dos casos de uso os elementos se relacionam, podendo existir relacionamentos:

- Entre atores:
- Entre atores e casos de uso
- Entre casos de uso

O tipo de relacionamento que será utilizado no diagrama de casos de uso do sistema estudado será entre atores e casos de uso, sendo representado pela seguinte ilustração:

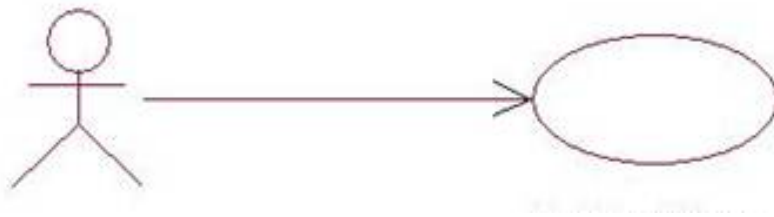


Figura 2. Representação gráfica do relacionamento entre ator e caso de uso.

A seta pode indicar que o ator é quem inicia o caso de uso, ou pode indicar apenas a direção dos dados. O balão é o caso de uso.

É importante saber que um caso de uso descreve um fluxo de eventos necessários para a realização de uma operação, e para que isso ocorra é preciso que existam pré e pós-condições que devem ser verdadeiras antes ou depois de o caso de uso ser executado.

Na seguinte página será apresentado o diagrama de caso de uso utilizado no desenvolvimento do programa do cartório.

6.1.1 Diagrama de casos de uso – aplicação

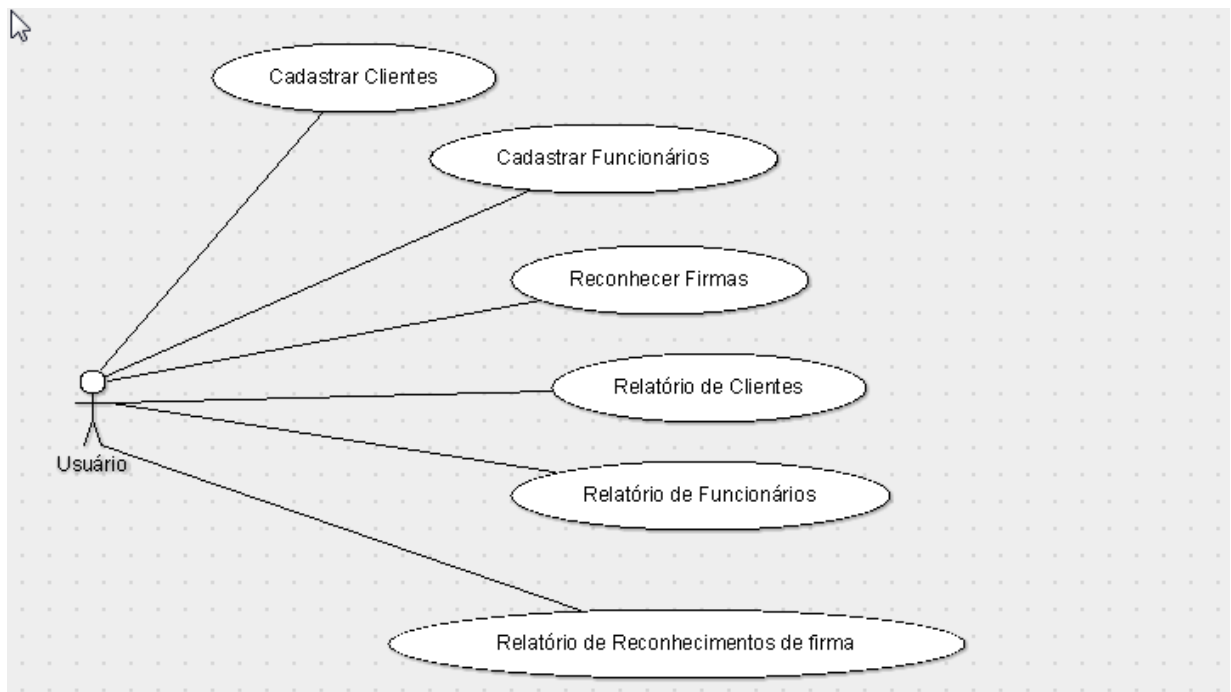


Diagrama 1 - Diagrama de Casos de Uso – Sistema Informatizado do Cartório.

Neste diagrama, o ator, denominado **Usuário 1** pode ser qualquer um dos funcionários do cartório. Mesmo que suas ações dependam do serviço que o cliente necessita, é o funcionário quem irá manipular o sistema, por isso ele é quem é o ator desses casos de uso.

Os casos de uso são as funções que o sistema do cartório precisa oferecer ao funcionário para que os serviços notariais sejam realizados de maneira correta e dinâmica.

6.2 Descrição de Cada um dos Casos de Uso do Sistema.

A seguir serão descritos os casos de uso do sistema informatizado do cartório. Primeiramente será apresentada uma descrição de como o serviço é feito, em seguida será mostrado como o sistema realiza essa tarefa, e posteriormente haverá a representação desse caso de uso com um ator.

6.2.1 Manter funcionários

Neste caso de uso, o **Usuário 1** poderá manter um cadastro atualizado de seus funcionários, como nome, endereço, telefone, estado civil, entre outros.



Figura 3. Caso de uso: Manter Funcionários.

6.2.2 Manter clientes

Para um cartório, é muito importante possuir um cadastro com os dados dos clientes que utilizam os serviços notariais. Essa importância é devida tanto pela agilidade que o cadastro proporciona quanto por questões de segurança e controle de atendimento.

Por isso, é necessário que o sistema também mantenha um registro de clientes, que deve conter alguns dados principais como nome completo, endereço, telefone de contato, estado civil, data de nascimento, número do RG e do CPF, entre outros. O **Usuário 1** é o responsável por coletar os dados de cada cliente e inseri-los no sistema, conforme demonstra a seguinte figura:

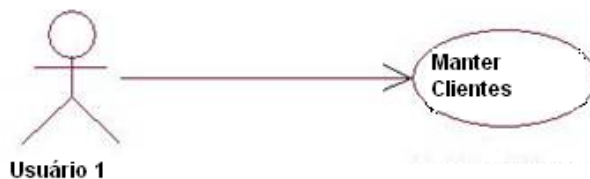


Figura 4. Caso de uso: Manter Clientes.

6.2.3 Reconhecer Firmas

Cada vez que um cliente utiliza os serviços notariais para que a firma de um documento seja reconhecida, o cartório disponibiliza a esse cliente um selo de

autenticidade. Esse selo, juntamente com uma etiqueta que contém o nome do funcionário que realizou o atendimento e a data em que o serviço foi realizado. Por motivos de segurança, é necessário que todos os reconhecimentos de firma sejam cadastrados no sistema.

Reconhecer firma é a comparação da assinatura do documento apresentado com a assinatura da ficha coletada anteriormente. Caso o cliente ainda não possua um cartão de assinatura, ele é feito imediatamente com a presença do dono da assinatura em questão.

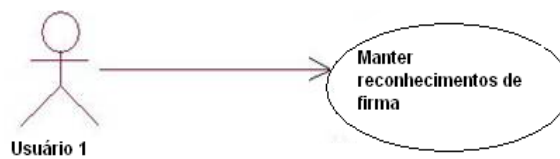


Figura 5. Caso de uso: Manter Reconhecimentos de Firma.

6.2.4 Relatório de reconhecimento de firmas

O relatório de reconhecimento de firmas facilita o controle sobre todos os reconhecimentos de firma feitos em um determinado período que será escolhido pelo **usuário 1**.

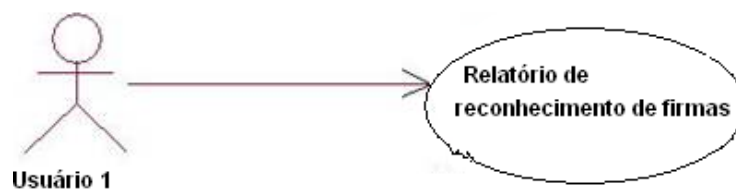


Figura 6. Caso de uso: Relatório de Reconhecimento de Firmas.

6.2.5 Relatório de clientes

O relatório clientes irá permitir que o **usuário 1** tenha acesso a todos os dados dos clientes que foram cadastrados em no cartório.

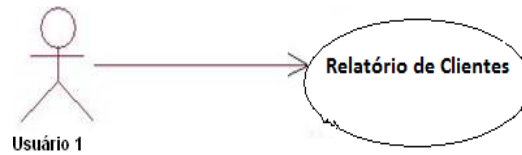


Figura 7. Caso de uso: Relatório de Clientes.

6.2.6 Relatório de funcionários

O relatório de Funcionários permitirá que o **Usuário 1** tenha acesso a todos ao registro de todos os funcionários do cartório.

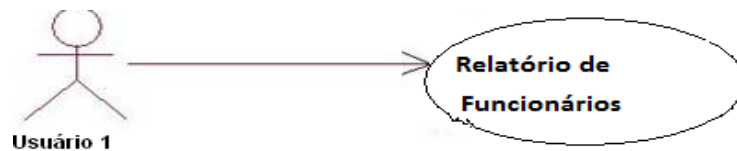


Figura 8. Caso de uso: Relatório de Cancelamento de Protestos.

6.3 Diagrama de Classes - Definição

O diagrama de classes é parte fundamental para que as pessoas envolvidas no projeto (usuários, clientes, desenvolvedores, etc.) compreendam tanto a estrutura do aplicativo quanto suas funcionalidades. O diagrama de classes é formado basicamente por:

- Classes
- Atributos da Classe
- Métodos da Classe

Classes são blocos de código que descrevem os objetos para o programa, são uma forma de abstrair um objeto do mundo real e transportá-lo para o programa, facilitando a compreensão e a forma de interpretação.

Atributos da Classe são as informações a respeito das classes. Essas informações estão contidas no sistema. Os atributos são modificados pelos métodos de classes, que através de regras alteram os atributos e o estado das classes. Os

métodos de Classes agem como um bloco de instruções que contém ou não regras capazes de modificar um objeto, ou de fazer com que uma classe interaja com outra classe. É importante lembrar que uma classe que não possui nenhum relacionamento não serve para nada, uma vez que a união ou associação das classes gera a execução de alguma das funcionalidades do sistema desenvolvido.

O diagrama de classes serve para que o programador tenha um controle visual sobre as funcionalidades do programa/sistema desenvolvido por ele. Alguns programadores acham irrelevante o uso dessa ferramenta, porém, quanto mais ferramentas puderem ser utilizadas para auxiliar o desenvolvimento de um sistema, menores serão as chances desse sistema apresentar falhas. Isso ocorre por que uma visão física do software pode apontar erros que passaram despercebidos pelos outros processos.

6.3.1 Diagrama de classes - Aplicação

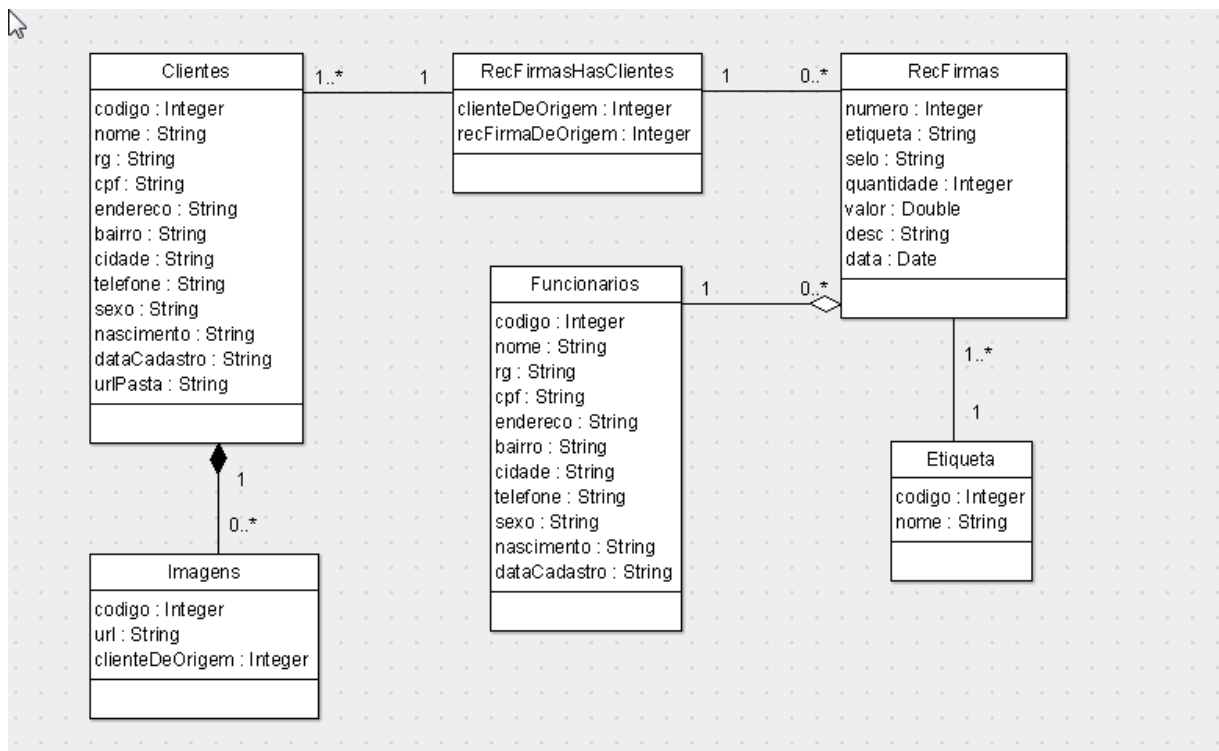


Diagrama 2 – Diagrama de Classes – Sistema Informatizado do Cartório

6.4 Diagrama de Entidade e Relacionamento – Definição

Outra ferramenta que auxilia o processo de programação de um sistema é o diagrama de entidade e relacionamento, mais conhecido como DER. Através desse diagrama, os elementos essenciais que foram extraídos nos processos de análise são estruturados de maneira sucinta.

Cada entidade (representada por um retângulo) é definida por atributos que são pertinentes ao sistema, quer dizer que as entidades são os elementos que armazenam dados, e esses dados são representados pelos atributos. Os relacionamentos entre as entidades são representados por losangos.

Em resumo, o diagrama de entidade e relacionamento descreve de forma geral a estrutura lógica de um banco de dados. Dessa maneira, tem-se as seguintes representações:

- Retângulos: Conjunto/entidade
- Elipses: Atributos.
- Losangos: Conjunto/relacionamento: Losangos.
- Linhas: Ligação entre atributos e conjuntos/entidade e conjuntos/entidade a conjuntos relacionamento.

O software DBDesigner auxilia no estudo dos diagramas de casos de uso pois, além de criar esse tipo de diagramas, ele gera um código SQL.

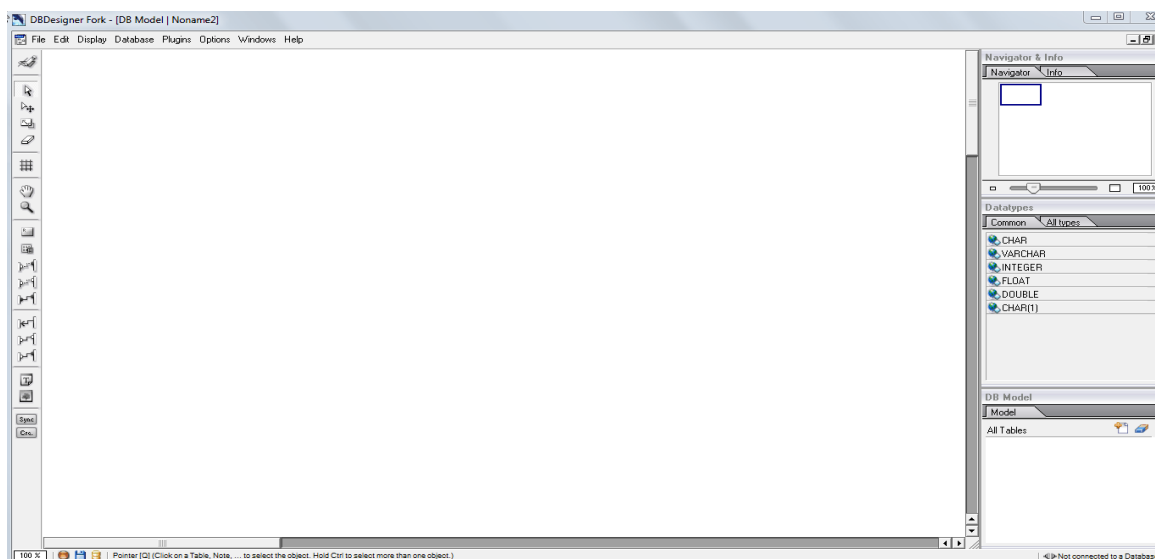


Figura 9. Print da tela do software DBDesigner.

6.4.1 – Diagrama de entidade e relacionamento - aplicação

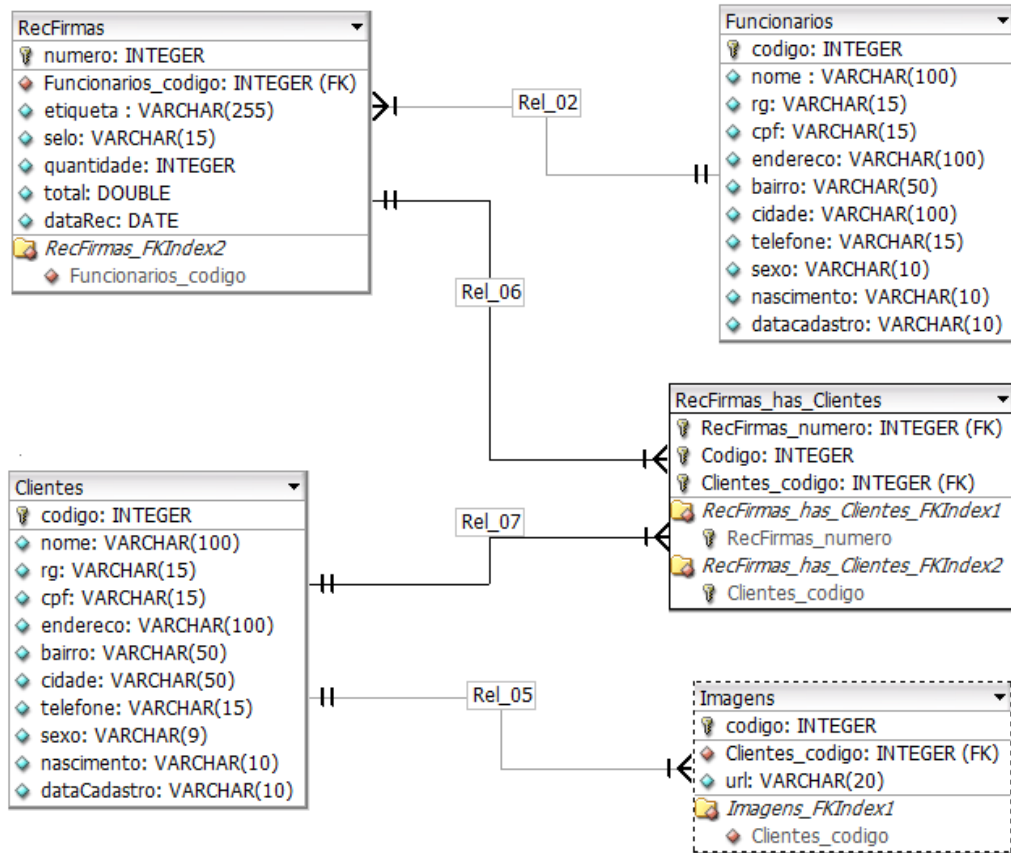


Diagrama 3 - Diagrama de entidade e relacionamento

6.5 Diagrama de Atividades – Definição

O diagrama de atividades tem por objetivo demonstrar o fluxo de atividades em um único processo. Ele mostra a interdependência entre cada uma das funções de um dado sistema.

Para confeccionar o diagrama de atividades abaixo, foi utilizada a ferramenta DBDesigner Fork.

6.5.1 Diagrama de Atividades – Aplicação

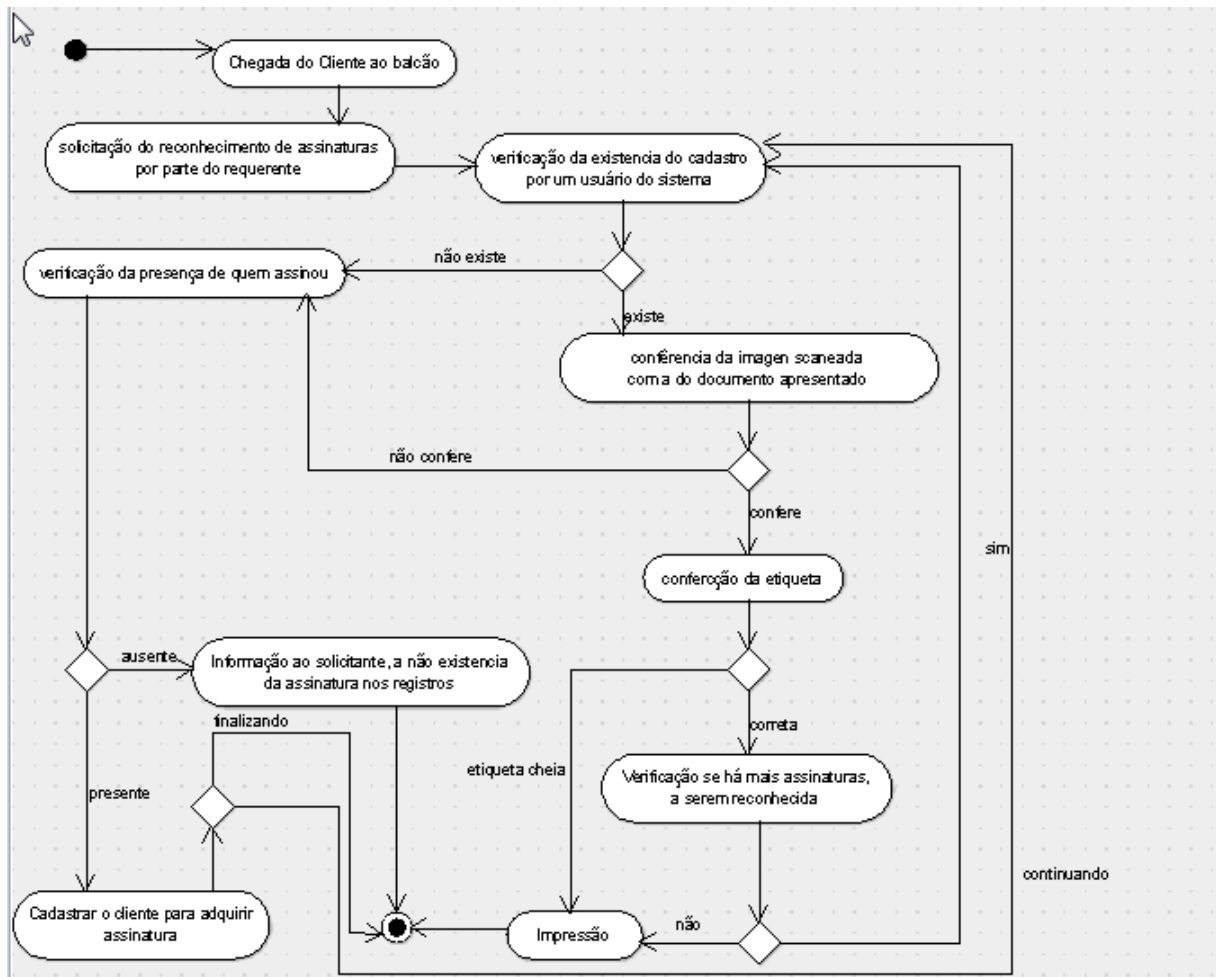


Diagrama 4 - Diagrama de atividades

7. TELAS DO SISTEMA

A seguir serão apresentadas as telas do sistema:

7.1 Tela inicial do sistema

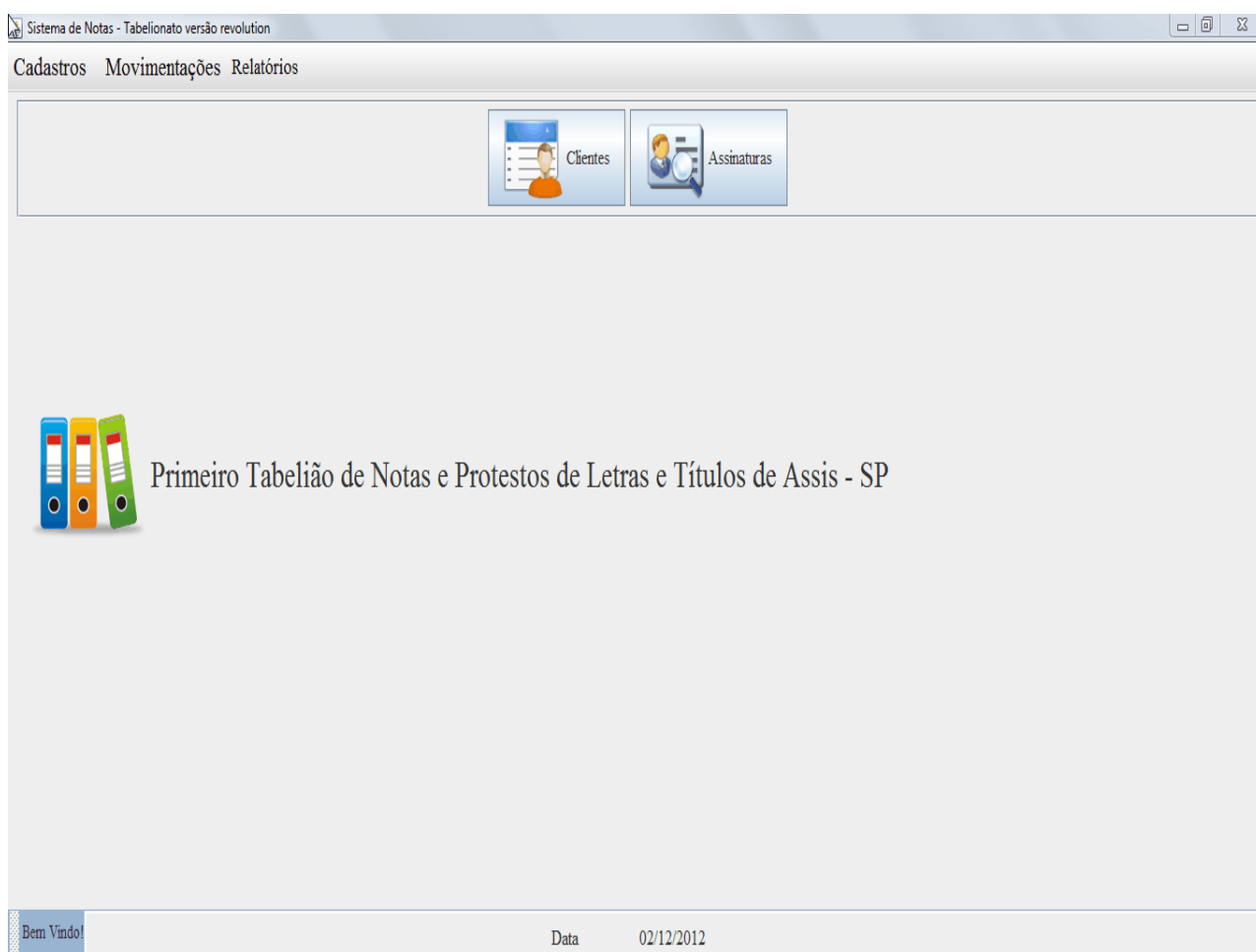


Figura 10 – Tela inicial do sistema

O usuário pode escolher entre os itens: Cadastro; Movimentações e Relatórios, além de poder pesquisar Clientes e Assinaturas.

7.2 Cadastro de funcionários

Sistema de Notas - Tabelionato versão revolution

Cadastros - Movimentações - Relatórios

Tabelião versão 1.0

CODIGO DATA DO CADASTRO

NOME

RG CPF ENDEREÇO

BAIRRO CIDADE TELEFONE () -

NASCIMENTO SEXO

ADMISSÃO DEMISSÃO

Opções

Gravar Alterar Excluir Novo Canelar Voltar

Bem Vindo! Data 02/12/2012

Figura 11 – Cadastro de funcionários

7.3 Cadastro de cliente

Sistema de Notas - Tabelionato versão revolution

Cadastros - Movimentações - Relatórios

Cadastro de Clientes Sistema Tabelião de Notas versão 1.0

Ficha Imagem

Código Nome

RG Emissão CPF Sexo

Endereço Bairro

Cidade Telefone () - Nascimento

Observações

Estado Civil Profissão

Menu

Gravar Alterar Excluir Novo Canelar Cartão Voltar

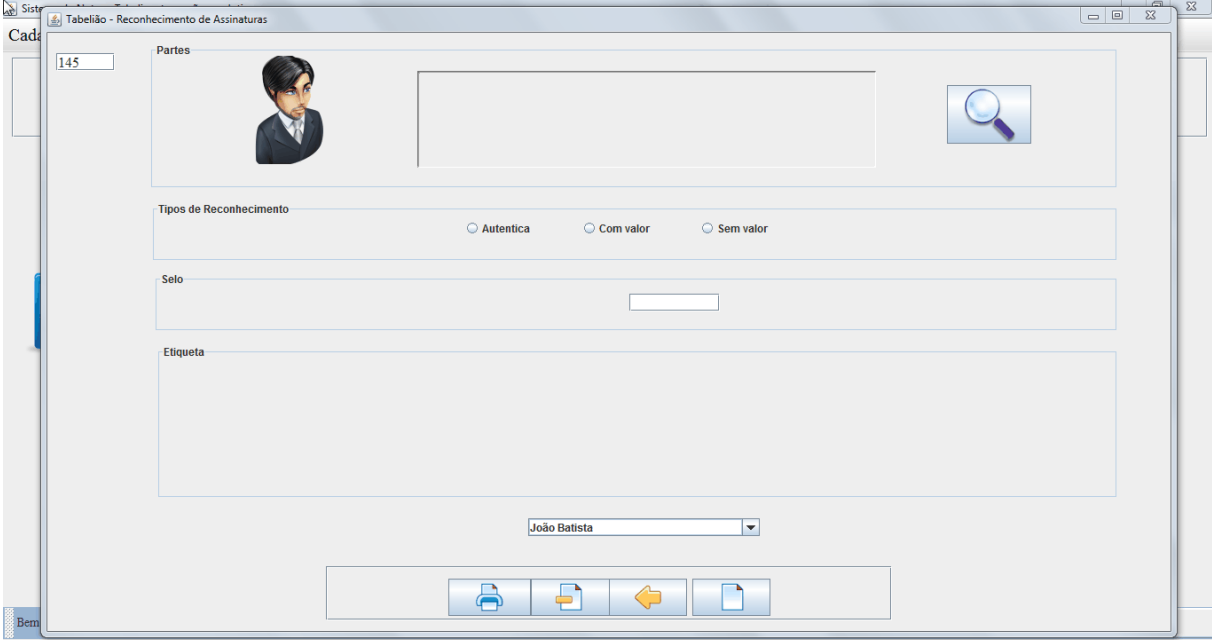
Clientes

02/12/2012

Bem Vindo! Data 02/12/2012

Figura 12 – Cadastro de cliente

7.4 Reconhecimento de assinaturas



145

Partes

Tipos de Reconhecimento

Autentica Com valor Sem valor

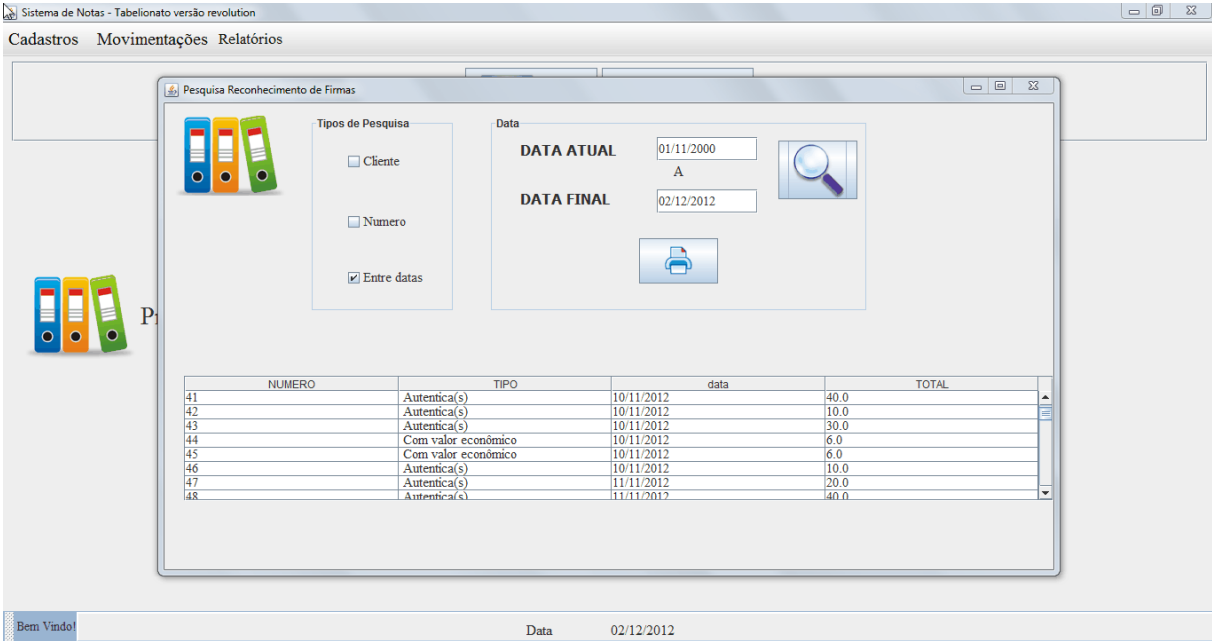
Selo

Etiqueta

João Batista

Figura 13 – Reconhecimento de assinaturas

7.5 Relatório de reconhecimento de firma



Tipos de Pesquisa

Cliente

Numero

Entre datas

Data

DATA ATUAL 01/11/2000

A

DATA FINAL 02/12/2012

NUMERO	TIPO	data	TOTAL
41	Autentica(s)	10/11/2012	40.0
42	Autentica(s)	10/11/2012	10.0
43	Autentica(s)	10/11/2012	30.0
44	Com valor econômico	10/11/2012	6.0
45	Com valor econômico	10/11/2012	6.0
46	Autentica(s)	10/11/2012	10.0
47	Autentica(s)	11/11/2012	20.0
48	Autentica(s)	11/11/2012	40.0

Bem Vindo! Data 02/12/2012

Figura 14 – Relatório de reconhecimento de firma

7.6 Relatório de clientes cadastrados

PRIMEIRO TABELIÃO DE NOTAS E PROTESTOS DE LETRAS E DE TÍTULOS DE ASSIS -SP
 RUA FLORIANO PEIXOTO 695 - CENTRO ASSIS - SP CEP 19800-011
 TELEFONE (18) 3302-1597 Domingo 02 Dezembro 2012

Relatório de Clientes

codigo	NOME	cpf	Cadastro
161	Bendito Moreira Pontes	011.705.469-00	2012-11-25
162	Caio Soares Costa	416.357.088-07	2012-11-25
163	Cristiano Aparecido Marcelino	265.017.918-83	2012-11-26
164	Daniel de Oliveira Soares	363.810.751-20	2012-11-25
165	Danilo Venuto Pereira	309.850.958-05	2012-11-26
166	Denise Cristina de Oliveira	217.765.858-24	2012-11-25
167	Denise Cristina de Oliveira	217.765.858-24	2012-11-25

Página 1 de 3

Figura 15 – Relatório de clientes cadastrados

7.7 Relatório de funcionários

PRIMEIRO TABELIÃO DE NOTAS E PROTESTOS DE LETRAS E DE TÍTULOS DE ASSIS -SP
 RUA FLORIANO PEIXOTO 695 - CENTRO ASSIS - SP CEP 19800-011
 TELEFONE (18) 3302-1597 Domingo 02 Dezembro 2012

Relatório de Funcionários

- 18 João Batista
- 19 José Américo Doiche Junior
- 20 Cristiano Salles Bechelli
- 21 Lourival Gama da Silva
- 22 Ana Maria de Souza Lencione
- 23 Alice Guerra da Silva
- 24 João Carlos
- 26 Ivone Rosa Ferreira Doiche
- 27 Sandra Doiche

Página 1 de 2

Figura 16 – Relatório de funcionários

CONSIDERAÇÕES FINAIS

As justificativas do sistema apontam o crescimento da informatização dos setores, em especial dos cartórios, foi programado um sistema que atende essa demanda, enfatizando o reconhecimento de firmas que é um dos serviços mais solicitados

Foram apresentadas três questões com relação à viabilidade do projeto: a possibilidade de criação de um software para ao Primeiro Cartório de Notas e Protestos de Assis; a possibilidade de oferecer um serviço ágil, seguro e de qualidade; e se os clientes se interessariam em usar esse sistema.

Os Conceitos Teóricos descrevem as teorias que envolveram tanto a elaboração da parte escrita do trabalho quanto a parte de programação do software. Pode-se dizer que é um dos capítulos mais importantes do trabalho, pois até mesmo os leitores que não estão familiarizados com as teorias do curso de Ciência da Computação poderão compreender os conceitos aqui abordados.

Os Métodos de Desenvolvimento estão voltados à descrição das ferramentas utilizadas na programação do software: Java como linguagem de programação, Eclipse IDE como ambiente de desenvolvimento, ArgoUML como modelador do desenho do software, MySQL como banco de dados e DBDesigner como modelador do banco de dados. É um capítulo cuja função é apresentar de maneira clara e rápida cada uma das ferramentas empregadas no desenvolvimento do software.

Depois é feita uma análise do sistema através de dados do cliente, e da coleta de requisitos, é baseado nesse capítulo que o sistema foi desenvolvido, uma vez que nele estão especificadas todas as funcionalidades do reconhecimento de firmas que o sistema deve oferecer a seu público alvo. Nesse capítulo foi definido que o sistema seria dividido em três partes: Cadastrar, Reconhecer firmas e Emitir Relatórios.

Em seguida são apresentados os diagramas que se fizeram necessários para a organização das funções do software: o diagrama de casos de uso, o diagrama de

classes e o diagrama de entidade e relacionamento e o diagrama de atividades, além da descrição de cada um dos casos de uso do sistema.

Por fim, pode-se concluir que o trabalho alcançou os objetivos propostos, criando um sistema para cartórios útil com interface intuitiva.

REFERÊNCIAS BIBLIOGRÁFICAS

Braude. E. Projeto de Software: Da programação à arquitetura: uma abordagem baseada em Java. São Paulo: Bookman, 2004.

Burnette. E. Eclipse IDE: guia de bolso. Trad. João Tortello. Porto Alegre: Bookman, 2006.

Cockburn, A. Escrevendo Casos de Uso Eficazes: um guia prático para desenvolvedores de software. Trad. Roberto Vedoato. Porto Alegre: Bookman, 2005.

Larman. C. Utilizando UML e Padrões: uma introdução à análise e ao projeto orientados a objeto e ao desenvolvimento iterativo. 3. ed. São Paulo. Bookman, 2005.

Martins, J.C.C. Gerenciando projetos de desenvolvimento de softwares com PMI, RUP e UML. 4. ed. Atual. Rio de Janeiro: Brasport, 2007.

Mattos. E. C. T. Programação de Softwares em Java. São Paulo: Digerate Books, 2007.

Paula Filho, W.P. Engenharia de Software: fundamentos, métodos e padrões. 3. ed. Rio de Janeiro: LCT, 2009.

Ramos, R. A. Treinamento Prático em UML. São Paulo: Digerate Books, 2006.