

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CÂMPUS CORNÉLIO PROCÓPIO
DIRETORIA DE GRADUAÇÃO E EDUCAÇÃO PROFISSIONAL
CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE
SISTEMAS

ENRIQUE CRUZ MACHADO

SISONG: SISTEMA GERENCIADOR DE ONGs

TRABALHO DE CONCLUSÃO DE CURSO

CORNÉLIO PROCÓPIO

2014

ENRIQUE CRUZ MACHADO

SISONG: SISTEMA GERENCIADOR DE ONGs

Trabalho de Conclusão de curso apresentado como requisito parcial para a obtenção do título de Bacharel, do curso de Análise e Desenvolvimento de Sistemas da Universidade Tecnológica Federal do Paraná – UTFPR-CP.

Orientador: Prof. André Luís dos Santos Domingues

CORNÉLIO PROCÓPIO

2014



Ministério da Educação
Universidade Tecnológica Federal do Paraná
Campus Cornélio Procópio
Nome da Diretoria
Nome da Coordenação
Nome do Curso



FOLHA DE APROVAÇÃO

“A Folha de Aprovação assinada”

À Jeová meu Deus e Senhor e a Jesus meu Salvador pela oportunidade de estar cursando em uma faculdade e pela força diária que me tem concedido para poder concretizar este trabalho.

À Valéria T. C. Machado, minha esposa e Mateus T. C. Machado, meu filho que estiveram todo o tempo ao meu lado apoiando e me incentivando a prosseguir.

À Armando Ávila Machado (in memoriam), meu pai e Judite Cruz Machado, minha mãe por todo o sacrifício e esforço que fizeram para que seus seis filhos não passassem pelas dificuldades que eles passaram.

Ao Pastor Décio M. Trindade por acreditar em mim e me incentivar a realizar o curso.

À Elaine C. M. Carmona, minha irmã pelo exemplo de dedicação, determinação, desprendimento e por estar junto a mim neste momento especial.

AGRADECIMENTOS

Agradeço Primeiramente a Deus por ter concedido esta oportunidade de estar realizando um curso superior e pela força, vigor e sabedoria que me tem dado durante todo o tempo, principalmente nas horas mais difíceis.

Também agradeço a minha família que me deu todo apoio necessário e tiveram paciência para comigo, os quais são a razão de todo meu esforço.

A todos os membros da minha igreja e de outras que intercederam por mim junto ao soberano Deus.

Ao meu orientador Prof. André Domingues, colaborou na elaboração do meu trabalho e me guiou nesta trajetória.

A Secretaria do Curso, pela cooperação.

Gostaria de deixar registrado também, o meu reconhecimento ao diretor Prof. Devanil Antônio Francisco e a todos que trabalham na UTFPR em todos os setores, que me apoiaram e ajudaram após o acidente que me foi acometido, pois acredito que sem o apoio que tive seria muito difícil vencer esse desafio.

Enfim, a todos os que por algum motivo contribuíram para a realização deste trabalho.

RESUMO

Cruz Machado, Enrique. **Sistema gerenciador de ONGs “SISONG”**. Trabalho de Conclusão de Curso (Graduação) – Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas pela Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2014.

Este trabalho apresenta o desenvolvimento de um sistema via web para o gerenciamento de uma ONG (Organização Não Governamental) que atua no 3º Setor. Esta entidade é sem fins lucrativos e tem como objetivo e finalidade o desenvolver ações voltadas ao bem comum através de arrecadações e distribuição de doações, palestras e outras atividades beneficentes, sendo composta pelo presidente, vice-presidente, secretários, tesoureiros e cooperadores. O sistema proposto visa organizar todas as informações possíveis de maneira mais dinâmica a fim de facilitar o gerenciamento de doações, cadastro de clientes e cadastro de doadores.

Palavras-chave: ONG (Organização não Governamental). Gerenciamento. Doações.

ABSTRACT

Cruz Machado, Enrique. **Sistema gerenciador de ONGs “SISONG”**. Trabalho de Conclusão de Curso (Graduação) – Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas pela Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2014.

This paper presents the development of a web-based system for managing an NGO (Non-Governmental Organization) that operates in the 3rd Sector. This organization is non-profit and aims to develop the purpose and actions for the common good through collections and distribution of donations, lectures and other charitable activities, being composed of the President, Vice President, secretaries, treasurers and cooperators. The proposed system aims to organize all possible information more dynamically to facilitate the management of donations, customer database and registry of donors.

Keywords: NGO (Non-Governmental Organization). Management. Donations.

LISTA DE FIGURAS

Figura 1 – Processo SCRUM.	15
Figura 2 – Caso de Uso.....	23
Figura 3 - Diagrama de Sequência Fazer Login	24
Figura 4 - Diagrama de Sequência Manter Atendente.....	25
Figura 5 - Diagrama de Sequência Manter Cliente.....	25
Figura 6 - Diagrama de Sequência Manter Atendimento.....	26
Figura 7 – Diagrama de Sequência Cadastrar Dependente	26
Figura 8 – Diagrama de Sequência Manter Doador	27
Figura 9 – Diagrama de Sequência Cadastrar Doação	27
Figura 10 – Diagrama de Entidade de Relacionamento (DER)	28
Figura 11 – Diagrama de Classe	29
Figura 12 – Diagrama de Classe (MVC).....	30
Figura 13 – Diagrama de Classe Atendente e Atendimento.....	31
Figura 14 – Diagrama de Classe Endereço.....	32
Figura 15 – Diagrama de Classe Doação.....	32
Figura 16 – Diagrama de Classe Cliente e Dependente.....	33
Figura 17 – Contagem de Pontos por Função.....	34
Figura 18 - Tela de Login.....	35
Figura 19 – Tela Lista Atendente.....	36
Figura 20 – Tela de Informações	37

LISTA DE SIGLAS

ONG	Organização Não Governamental
API	Interface de Programação de Aplicações
RUP	Rational Unified Process
EUP	Enterprise Unified Process
XP	Programação Extrema
SQL	Structured Query Language
ANSI	American National Standards Institute
JVM	Máquina Virtual Java
HTML	Linguagem de Marcação de Hipertexto
CSS	Cascading Style Sheets (folha de estilos em cascata)
JSF	Java Server Faces
DER	Diagrama de Entidade de Relacionamento
ALI	Arquivo Lógico Interno
AIE	Arquivo de Interface Externa
IE	Input Externo (entrada externa)
OE	Output Externo (saída externa)
MVC	Model-view-controller (modelo-visão-controlador)

SUMÁRIO

1 INTRODUÇÃO	11
2 JUSTIFICATIVA	12
3 OBJETIVOS	13
3.1 Objetivos Gerais	13
3.2 Objetivos Específicos	13
4. METODOLOGIA E TECNOLOGIA UTILIZADA	14
4.1 Processo de Desenvolvimento	14
4.2 MVC (Model View-Controller)	15
4.3 Banco de dados PostgreSql 9.1	16
4.4 Linguagem de Programação	16
4.5 NetBeans IDE 7.4.....	17
4.6 PrimeFaces 4.0	17
4.7 Bootstrap.....	17
4.8 Google Maps API	18
4.9 Padrões de Testes	18
4.9.1 Teste de Registros	18
4.9.2 Teste De Acesso	19
4.9.3 Testes de Usabilidade	19
5. DESENVOLVIMENTO.....	21
5.1 Cronograma Inicial	21
5.2 Cronograma Final.....	22
5.3 Caso de Uso	23
5.4 Priorização dos Requisitos	23
5.5 Diagrama de Sequência	24
5.5.1 Diagrama de Sequência Fazer Login	24
5.5.2 Diagrama de Sequência Manter Atendente	25
5.5.3 Diagrama de Sequência Manter Cliente	25
5.5.4 Diagrama de Sequência Manter Atendimento	26
5.5.5 Diagrama de Sequência Cadastrar Dependente	26
5.5.6 Diagrama de Sequência Manter Doador.....	27
5.5.7 Diagrama de Sequência Cadastrar Doação	27
5.6 Diagrama de Entidade de Relacionamento (DER).....	28
5.7 Diagrama de Classes	29

5.8 Diagrama de Classes(MVC)	30
5.9 Pontos por Função	34
5.10 Telas	35
6. PRINCIPAIS DIFICULDADES	39
7. CONSIDERAÇÕES FINAIS	40

1 INTRODUÇÃO

Em todo o território nacional e em outros países existem ONGs as organizações não governamentais, sendo um grupo social organizado, sem fins lucrativos que atuam no terceiro setor da sociedade civil. Caracterizam-se por ações de solidariedade pelas políticas públicas atuando em diversas áreas, tais como: Meio ambiente, combate à pobreza, assistência social, saúde, educação, reciclagem, desenvolvimento sustentável, entre outras. Essas organizações podem complementar o trabalho do Estado, podendo receber financiamentos e doações do mesmo, assim como de entidades privadas, também obtém recursos através de vendas de produtos e da população em geral (através de doações). Grande parte da mão-de-obra que atua nas ONGs é formada por voluntários. Possuem funções importantes na sociedade, pois seus serviços chegam a locais e situações em que o Estado é pouco presente. As ONGs não têm valor jurídico. No Brasil, três figuras jurídicas correspondentes no novo Código Civil compõem o terceiro setor: associações, fundações e organizações religiosas (SEMAD, 2014).

A finalidade deste documento é apresentar o desenvolvimento de um sistema via Web para atender às necessidades de uma ONG, denominada ONG ORAR, que está situada em Cornélio Procópio na Rua Portugal Nº302, Bairro Centro onde são desenvolvidos trabalhos de ação social e assistencialismo a pessoas carentes. Para atingir o objetivo proposto o trabalho apresenta nesta seção a introdução contextualizando a situação observada, na seção 2 a justificativa, na seção 3 os objetivos, na seção 4 a metodologia e a tecnologia utilizada, na seção 5 o desenvolvimento, na seção 6 as principais dificuldades e a seção 7 apresenta as considerações finais do trabalho.

2 JUSTIFICATIVA

A entidade recebe diferentes tipos de doações e em quantidades variadas. O desenvolvimento do sistema ajudará o trabalho de organização dessas doações e distribuição das mesmas, também possibilitará apresentar relatórios precisos para prestações de contas.

Foi realizada uma pesquisa dos softwares existentes no mercado e verificou-se que os existentes, são poucos conhecidos e difere muito uns dos outros, a maioria são complexos demais, porém não atendem a necessidade da ONG em questão. Como cada organização não governamental tem suas peculiaridades e atua de forma diferente, a maioria dos softwares torna-se inviável. O software proposto tem a finalidade de atender às necessidades de um cliente específico, no caso, a ONG ORAR sanando suas dificuldades.

O quadro abaixo mostra o resultado da pesquisa realizada.

Software	Principais características
Master Manager	Módulos do sistema, Básico – Financeiro Básico – Projetos, Acadêmico, Atendimento, Contabilidade, Compras, Patrimônio, Licitação, Recursos humanos, Arquivo virtual, Protocolo, Acesso via Web, Relatórios.
ONG Fácil	Projetos: Cadastro do projeto com as informações gerais, cronograma, atividades e orçamento. Cadastro de Doadores, Beneficiários e Recurso de Pessoal. Controle de Frequência: acompanhamento da presença dos beneficiários nas atividades envolvidas. Doações: Controle das doações realizadas.
KAD ONG	Cadastros Consultas: tarefas, eventos, viagens, projetos, estoque. Diversos: compras, anotações, agenda, promissória avulso, recibo avulso.
SISONG	Cadastros de usuário, cliente, doador e doação. Registro de atendimentos disponibilizados. Relatórios. Mapeamento de locais com maior concentração de pessoas necessitadas.

Quadro 1- Pesquisa de Softwares

Fonte: Autoria Própria

3 OBJETIVOS

O Sistema SISONG tem o objetivo de realizar o gerenciamento das doações recebidas pela entidade, o cadastro de clientes (beneficiários), o cadastro de doadores, dos atendimentos realizados e mapear os locais onde há maior concentração de pessoas necessitadas a fim de apresentar algumas estatísticas para melhorar o serviço prestado pela entidade, facilitando assim o controle e gerando informações que irão ajudar na prestação de contas junto aos órgãos competentes a fim de angariar recursos para a entidade.

3.1 Objetivos Gerais

O objetivo deste projeto é desenvolver um software que gerencie a administração da ONG informatizando os cadastros e atendimentos, visando a agilidade nas tomadas de decisões e facilitando o controle dos trabalhos desenvolvidos podendo ser acessado via web.

3.2 Objetivos Específicos

Os objetivos específicos do sistema são:

- Manter os registros de: clientes (beneficiários); doações; doadores atendimentos realizados.
- Disponibilizar informações sobre os trabalhos realizados.

4. METODOLOGIA E TECNOLOGIA UTILIZADA

Nos tópicos seguintes será apresentada a metodologia e as tecnologias utilizadas para o desenvolvimento do sistema proposto, as quais seguirão a seguinte ordem: Processo de desenvolvimento Scrum, banco de dados Postgres, linguagem de programação Java, ambiente de desenvolvimento NetBeans, Frameworks PrimeFaces e Bootstrap e a API do Google Maps.

4.1 Processo de Desenvolvimento

Um dos principais mecanismos para obter um software de qualidade é o processo de desenvolvimento que consiste em um conjunto de atividades, parcialmente ordenadas com a finalidade de obter o produto de software. Existem várias metodologias utilizadas hoje em dia, como por exemplo: Análise estruturada, orientada a objeto, RUP (Rational Unified Process), EUP (enterprise Unified Process), Scrum, XP (Programação Extrema) entre outras (PRESSMAN, 2006).

Um dos processos analisados foi o desenvolvimento ágil SCRUM (KNIBERG, 2007) que trabalha com equipes pequenas, para maximizar o compartilhamento de informação, o processo possui incrementos contínuos no software que podem ser inspecionados, testados documentados e expandidos, pois segundo (PRESSMAN, 2006) os testes e a documentação são feitos à medida que o software é construído.

Para o desenvolvimento escolheu-se a metodologia ágil de desenvolvimento Scrum Solo, por melhor se adequar ao projeto em questão, tendo em vista que, trata-se de um único desenvolvedor, este processo é uma versão adaptada do Scrum para programadores solo que utiliza alguns princípios do Scrum como: um backlog de produto (são histórias pendentes), um backlog de Sprint (resume as tarefas que são feitas no decorrer do desenvolvimento), um Sprint (um período de tempo que resulta em parte do software pronto) e uma retrospectiva de Sprint (Henrik KNIBERG, 2007).

A figura1 abaixo mostra o exemplo do ciclo de vida do processo Scrum.



Figura 1 – Processo SCRUM.

Fonte: < <http://www.sstecnologia.com.br/> >

4.2 MVC (Model View-Controller)

MVC (Model View-Controller) é uma abordagem composta por três objetos: Modelo, Visão e o Controlador que são separados por meio de um protocolo do tipo inserção/ notificação entre eles aumentando a flexibilidade de reutilização de código (GAMMA, 2006, p.20). O Modelo implementa o objeto da aplicação, a Visão é a apresentação da tela para o usuário e o Controlador define a reação da interface de acordo com as entradas do usuário. O objetivo da visão é garantir o reflexo do estado do Modelo. Toda vez que os dados do Modelo são alterados ele notifica as visões que dependem dele e em resposta cada visão tem a oportunidade de se atualizar. Este tipo de abordagem permite que um modelo possua múltiplas visões sem precisar reescrevê-lo.

A resposta em um objeto Controlador é encapsulada pela MVC. Existe uma hierarquia de controladores, o que permite a criação de novos controladores como variações de um controlador já existentes. Uma visão utiliza uma subclasse de controlador para construir uma estratégia para respostas diferentes e apenas substitui a instância por outro controlador. Em tempo de execução é possível alterar o controlador de uma visão para dar outro tipo de resposta ao usuário.

4.3 Banco de dados PostgreSql 9.1

Dentre os diversos bancos de dados existentes foi escolhido para o projeto o banco de dados PostgreSQL, o motivo principal da escolha foi por ele ser de código aberto e fazer parte de um projeto produzido recentemente onde foi possível ter uma maior familiaridade com o banco.

O PostgreSQL é um sistema de gerenciamento de banco de dados relacional, suporta grande parte do padrão SQL, oferece muitas características modernas como: consultas complexas, chaves estrangeiras, gatilhos, visualizações, integridade transacional e controle de concorrência (Pereira Neto, 2003).

Por possuir sua licença livre pode ser usado, modificado ou distribuído por qualquer pessoa gratuitamente para qualquer finalidade, seja privada, comercial ou acadêmico. Álvaro Pereira Neto no livro PostgreSQL diz:

“o PostgreSQL é Open Source (Código Aberto) e implementam os padrões SQL ANSI 92,96,99 (este último, englobando definições originárias da Orientação a Objetos)”.

4.4 Linguagem de Programação

A linguagem escolhida para o projeto foi a linguagem de programação Java, o motivo principal da escolha foi aproveitar o conhecimento obtido em um projeto desenvolvido na empresa Engine Software. A linguagem foi apresentada oficialmente pela Sun (Oracle, 2014) em 23 de maio de 1995, sua arquitetura baseia-se na existência de uma Máquina Virtual Java (JVM) que deixa transparente para a aplicação a existência do Sistema operacional e do Hardware sua função é carregar de forma segura todas as classes, verificar se os bytecodes são compatíveis e não violam a integridade de segurança do sistema e interpretar o código para a plataforma desejada.

Será utilizada também Java Script que é a linguagem de script da Web, que vem sendo utilizada em todas as páginas HTML (Linguagem de Marcação de Hipertexto) hoje em dia (Valdés, 2007).

4.5 NetBeans IDE 7.4

Devido a experiência obtida em projetos anteriores durante a realização do curso, optou-se pela utilização do NetBeans 7.4 como ambiente de desenvolvimento a ser empregado no trabalho. O NetBeans permite desenvolver rapidamente programas para desktop, dispositivos móveis e aplicações web, bem como aplicações HTML5 com HTML, Java Script e CSS. O IDE também fornece um grande conjunto de ferramentas para desenvolvedores PHP e C / C ++, sua licença é livre (código aberto) e possui uma grande comunidade de usuários e desenvolvedores em todo o mundo (Oracle, 2013).

4.6 PrimeFaces 4.0

Com o intuito de aprender e testar novas tecnologias, foi inserido o PrimeFaces, que é uma fonte de conjunto de componentes JSF com várias extensões dentre elas o Built-in Ajax com base no padrão JSF 2.0 APIs Ajax, kit móvel UI para criar aplicações web móveis para dispositivos portáteis, skin com mais de 35 temas internos com suporte para ferramenta visual de tema designer e muitas outras. Possui também um rico conjunto de componentes de HTML editor, diálogo, auto complete, gráficos, uma extensa documentação e uma grande comunidade de usuários todos estes dados podem ser encontrado e baixados no ambiente virtual do PrimeFaces (CIVICI, 2014).

4.7 Bootstrap

Conforme informações contidas no site oficial, Bootstrap é a estrutura mais popular de HTML, CSS e JS para o desenvolvimento ágil em projetos na web, ele é feito para pessoas de todos os níveis, dispositivos de todas as formas, e

projetos de todos os tamanhos, têm como objetivo tornar mais fácil o desenvolvimento de interface (front-end.) para páginas web disponibilizando padrões para os elementos HTML mais usados, além de elementos personalizados, sendo a última tendência na web (Otto, 2014).

4.8 Google Maps API

Para mapear os locais onde residem a maior parte dos beneficiários e adicionar informações ao mapa, foi utilizado a API (Application Programming Interface ou Interface de Programação de Aplicação) que é a forma de comunicação com funções pré-programadas e definidas pelo fornecedor, no caso, será utilizado o Google como fornecedor, conforme o Google “A API do Google Maps permite usar Java Script para incorporar o Google Maps em sua página da web. A API fornece diversos utilitários para manipular mapas...”, é possível também, adicionar conteúdos ao mapa através de serviços oferecidos pela API (Google Maps, 2014).

4.9 Padrões de Testes

Para testar o projeto foram realizados testes de entrada de dados. Abaixo segue alguns testes realizados:

4.9.1 Teste de Registros

Caso de Teste: Realizar o cadastro da doação informando os dados via interface.

Resultado Esperado: Registrar o cadastro da doação

Resultado Obtido: Cadastro das doações foi registrado com sucesso

Caso de Teste: Manter Doador

Resultado Esperado: Registrar e alterar dados dos doadores

Resultado Obtido: Doador registrado e alterado com sucesso

Caso de Teste: Manter Cliente

Resultado Esperado: Registrar cliente realizar alterações

Resultado Obtido: Cliente registrado e as alterações realizadas com sucesso

4.9.2 Teste De Acesso

Caso de Teste: Doador

Resultado Esperado: Acessar as informações do cadastro de doador

Resultado Obtido: Informações acessadas com sucesso

Caso de Teste: Cliente

Resultado Esperado: acessar cadastro de Cliente

Resultado Obtido: cadastro acessado com sucesso

Caso de Teste: Doação

Resultado Esperado: acessar cadastro de Doação

Resultado Obtido: cadastro acessado

4.9.3 Testes de Usabilidade

Foram realizados testes de usabilidade do software com informações obtidas em ErgoList que utilizam alguns critérios onde determinam a ergonomia de uma interface homem-computador. (Cybis, 2011).

Critério 1 : Presteza

Pergunta: Todos os campos e mostradores de dados possuem rótulos identificativos?

Resposta: Sim

Situação analisada: Cadastros de Doadores

Critério 2 : Legibilidade

Pergunta: Os grupos de objetos de controle e de apresentação que compõem as caixas de diálogo e outros objetos compostos encontram-se alinhados vertical e horizontalmente?

Resposta: Sim

Situação analisada: Cadastros de Clientes

Critério 3 : Densidade informacional

Pergunta: As telas apresentam somente os dados e informações necessários e indispensáveis para o usuário em sua tarefa?

Resposta: Sim

Situação analisada: Cadastros de Doação

Critério 4 : Correção de erros

Pergunta: Depois de um erro de digitação de um comando ou de dados, o usuário tem a possibilidade de corrigir somente a parte dos dados ou do comando que está errada?

Resposta: Sim

Situação analisada: Cadastro de Cliente

5. DESENVOLVIMENTO

O quadro abaixo apresenta o cronograma inicial do projeto previsto originalmente, na subseção 5.2 e apresentado o cronograma final reajustado devido as dificuldades encontradas que ocasionaram a falha no prazo de entrega.

5.1 Cronograma Inicial

Nome da tarefa	Duração	Início	Término
Projeto SisOng	157 dias	Sex 08/11/13	Ter 17/06/14
Definição do projeto	31 dias	Sex 29/11/13	Sex 10/01/14
Levantamento de requisitos	13 dias	Sex 29/11/13	Ter 17/12/13
Descritivo do sistema	4 dias	Ter 17/12/13	Sex 20/12/13
Diagrama de Caso de Uso	4 dias	Seg 23/12/13	Qui 26/12/13
Diagrama de Classe	3 dias	Qui 26/12/13	Seg 30/12/13
Criar Plano do Projeto	5 dias	Seg 30/12/13	Sex 03/01/14
Análise	5 dias	Seg 06/01/14	Sex 10/01/14
Sprint 01	30 dias	Seg 06/01/14	Sex 14/02/14
Analisar requisitos e documentar	6 dias	Seg 06/01/14	Seg 13/01/14
Identificar tecnologias	3 dias	Ter 14/01/14	Qui 16/01/14
Iniciar a implementação do Banco	7 dias	Qui 16/01/14	Sex 24/01/14
Implementação do cliente, atendente e endereço	5 dias	Seg 27/01/14	Sex 31/01/14
Protótipo de interface do cliente, atendente e endereço	3 dias	Seg 03/02/14	Qua 05/02/14
Testes	3 dias	Qua 05/02/14	Sex 07/02/14
Documentar	5 dias	Seg 10/02/14	Sex 14/02/14
Sprint 02	30 dias	Seg 17/02/14	Sex 28/03/14
Analisar requisitos	3 dias	Seg 17/02/14	Qua 19/02/14
Atualizar Diagramas	2 dias	Qua 19/02/14	Qui 20/02/14
Implementação do doador	15 dias	Qui 20/02/14	Qua 12/03/14
Protótipo de interface do doador	19 dias	Qui 13/02/14	Ter 11/03/14
Teste	3 dias	Qua 12/03/14	Sex 14/03/14
Alterações necessárias do processo	9 dias	Ter 18/03/14	Sex 28/03/14
Sprint 03	31 dias	Seg 07/04/14	Seg 19/05/14
Analisar requisitos e documentar	3 dias	Seg 07/04/14	Qua 09/04/14
Atualizar Diagramas	2 dias	Qui 10/04/14	Sex 11/04/14
Implementação da doação	9 dias	Seg 14/04/14	Qui 24/04/14
Protótipo de interface da doação	12 dias	Sex 25/04/14	Seg 12/05/14
Testes	3 dias	Ter 13/05/14	Qui 15/05/14
Documentar	2 dias	Sex 16/05/14	Seg 19/05/14
Ajustes Finais	10 dias	Ter 20/05/14	Seg 02/06/14
Apresentação	1 dia	Qua 11/06/14	Qua 11/06/14

Quadro 1 – Cronograma Inicial
Fonte: Autoria Própria

5.2 Cronograma Final

Nome da tarefa	Duração	Início	Término
Projeto SisOng	99 dias	Ter 22/07/14	Sex 05/12/14
Redefinição do projeto	9 dias	Ter 22/07/14	Seg 04/08/14
Análise dos requisitos	5 dias	Ter 22/07/14	Seg 28/07/14
Priorização dos requisitos	4 dias	Ter 29/07/14	Qui 31/08/14
Sprint 01	30 dias	Seg 04/08/14	Sex 12/09/14
Analisar e documentar	1 dia	Seg 04/08/14	Seg 04/08/14
Atualizar Diagramas	3 dias	Ter 05/08/14	Qui 07/08/14
Implementação do atendimento	7 dias	Seg 11/08/14	Ter 19/08/14
Implementar no banco o atendimento	5 dias	Qua 20/08/14	Ter 26/08/14
Protótipo de interface atendimento	3 dias	Qua 27/08/14	Sex 29/08/14
Testes	3 dias	Seg 01/09/14	Qua 03/09/14
Documentar	5 dias	Qui 04/09/14	Qua 10/09/14
Sprint 02	28 dias	Seg 15/09/14	Qua 22/10/14
Analisar requisitos	3 dias	Seg 15/09/14	Qua 17/09/14
Desenvolvimento	8 dias	Seg 22/09/14	Qua 01/10/14
Implementação do dependente	8 dias	Qua 01/10/14	Sex 10/10/14
Teste	3 dias	Seg 13/10/14	Qua 15/10/14
Alterações necessárias do processo	4 dias	Qui 16/10/14	Ter 21/10/14
Sprint 03	21 dias	Qui 23/10/14	Qui 20/11/14
Analisar requisitos e documentar	3 dias	Qui 23/10/14	Seg 27/10/14
Implementação dos recursos adicionais	12 dias	Ter 28/10/14	Qua 12/11/14
Testes	1 dia	Qui 13/11/14	Qui 13/11/14
Documentar	4 dias	Qui 13/11/14	Ter 18/11/14
Ajustes Finais	7 dias	Qui 20/11/14	Sex 28/11/14
Apresentação	5 dias	Seg 01/12/14	Sex 05/12/14

Quadro 2 – Cronograma Final

Fonte: Autoria Própria

5.3 Caso de Uso

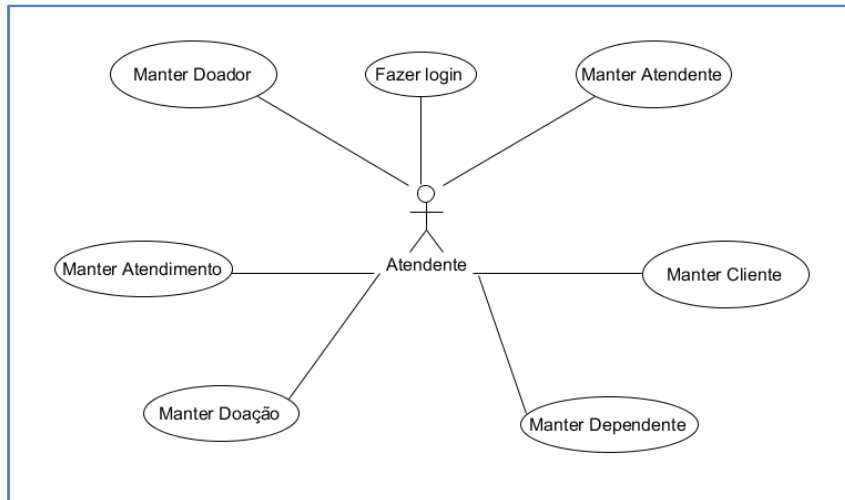


Figura 2 – Caso de Uso
Fonte: Autoria Própria

5.4 Priorização dos Requisitos

Nº	Funcionalidade	Prioridade do Cliente	Entrega
1	Manter Doação	Crítico	1
2	Manter Clientes	Crítico	1
3	Manter Doadores	Crítico	1
4	Manter Atendente	Crítico	1
5	Manter Atendimento	Crítico	
6	Fazer Login	Necessário	2
7	Emitir Relatórios	Útil	3

Segue a lista de alguns requisitos que foram surgindo ao longo do projeto e ficou fora do cronograma final devido o tempo que seria necessário para o desenvolvimento não ser o suficiente.

Nº	Funcionalidade	Prioridade do Cliente
1	Manter Entidades	Desejável
2	Manter Parcerias	Desejável
3	Manter Lista de Pedidos	Desejável
4	Manter Atividades desenvolvidas	Desejável

5.5 Diagrama de Sequência

Após definir qual o papel do sistema através dos casos de uso, foi definido como o software realizará seu papel utilizando sequências de operações que são demonstradas abaixo.

5.5.1 Diagrama de Sequência Fazer Login

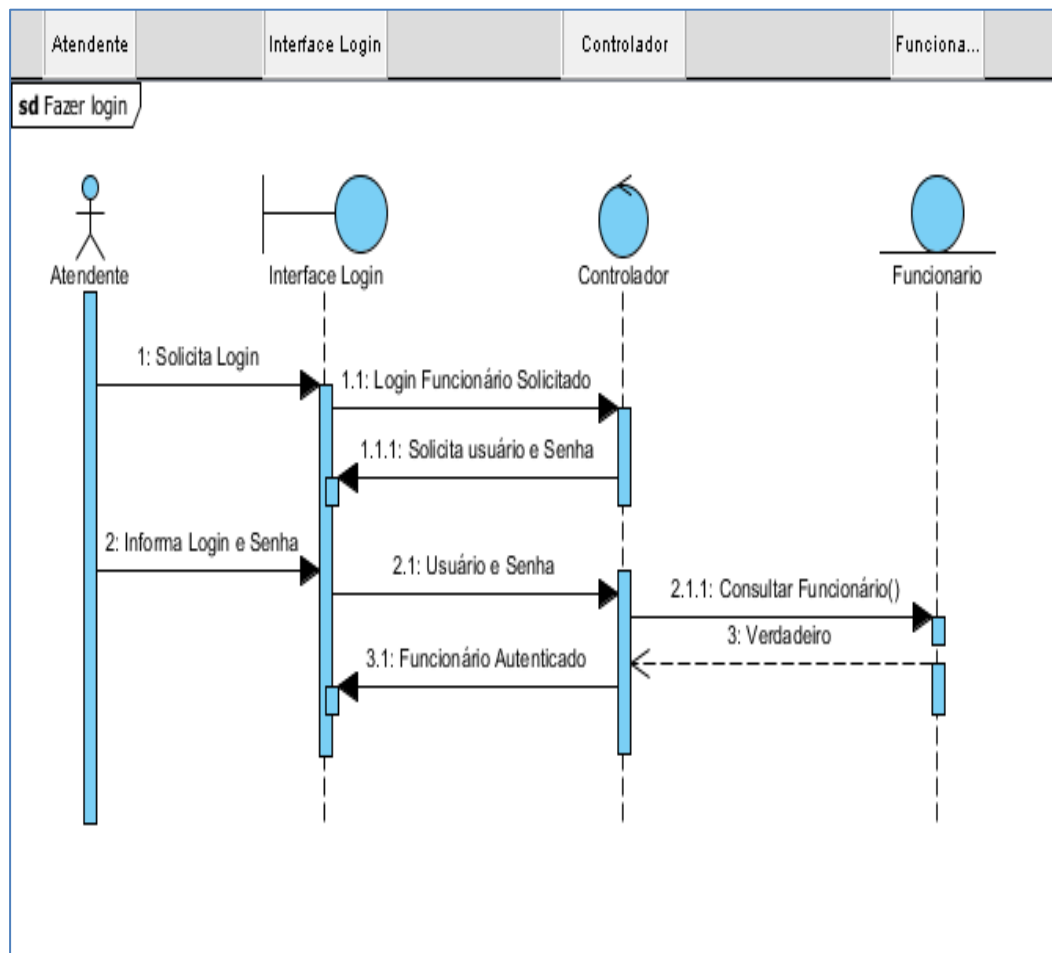


Figura 3 - Diagrama de Sequência Fazer Login
Fonte: Autoria Própria

5.5.2 Diagrama de Sequência Manter Atendente

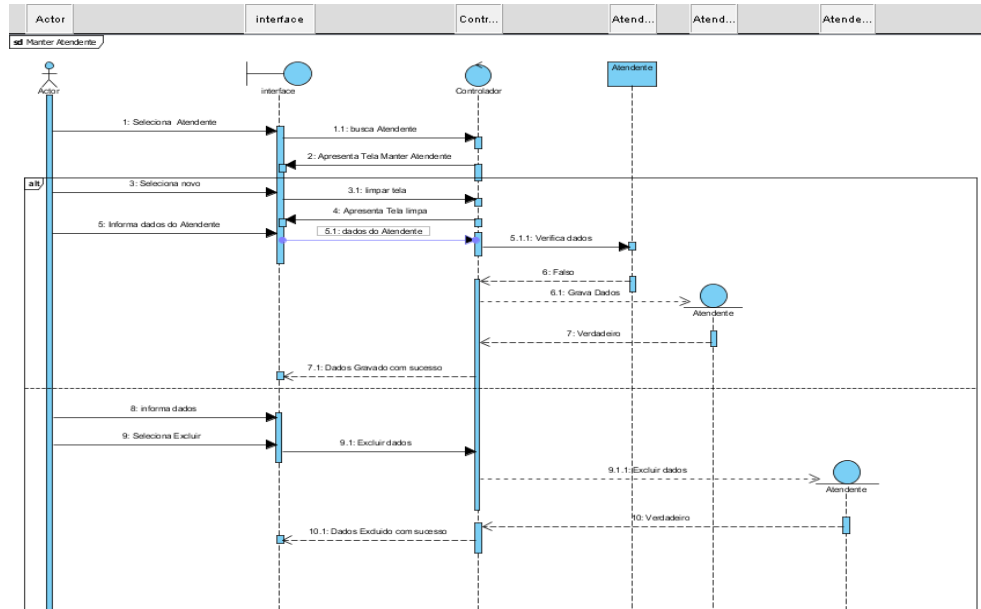


Figura 4 - Diagrama de Sequência Manter Atendente
 Fonte: Autoria Própria

5.5.3 Diagrama de Sequência Manter Cliente

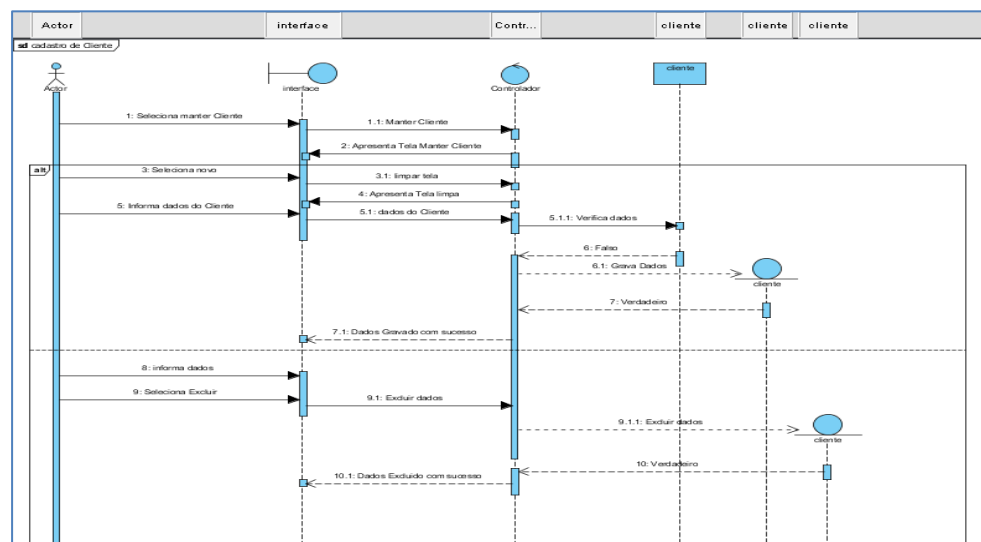


Figura 5 - Diagrama de Sequência Manter Cliente
 Fonte: Autoria Própria

5.5.4 Diagrama de Sequência Manter Atendimento

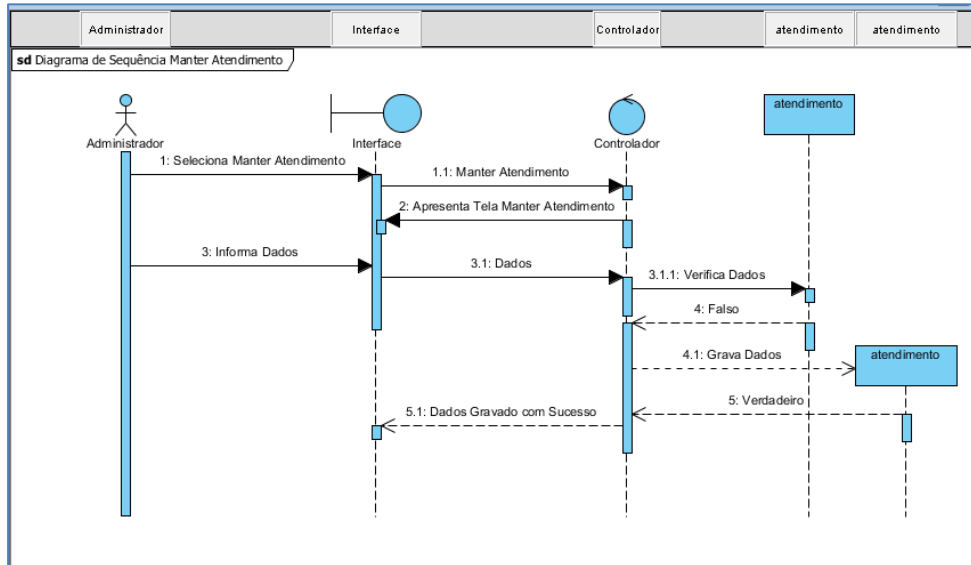


Figura 6 - Diagrama de Sequência Manter Atendimento
 Fonte: Autoria Própria

5.5.5 Diagrama de Sequência Cadastrar Dependente

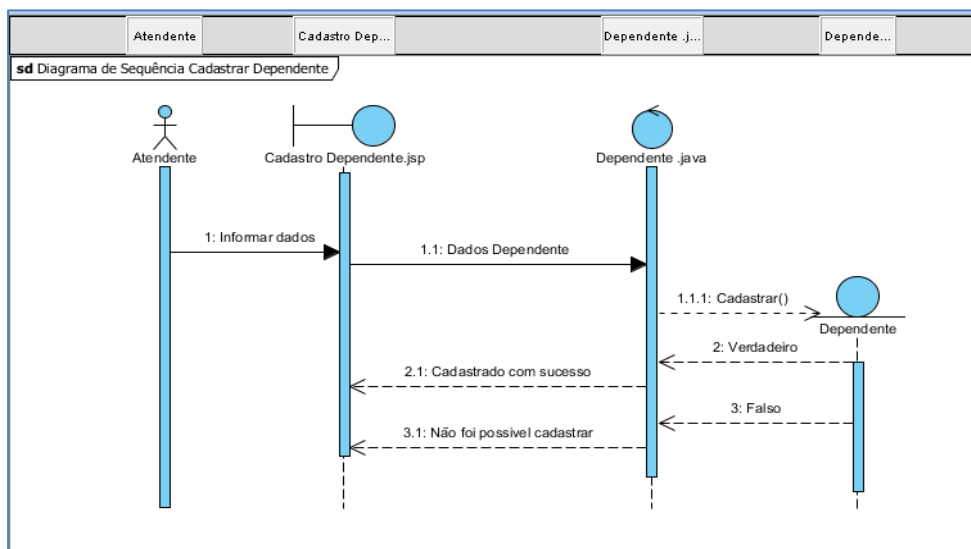


Figura 7 – Diagrama de Sequência Cadastrar Dependente
 Fonte: Autoria Própria

5.5.6 Diagrama de Sequência Manter Doador

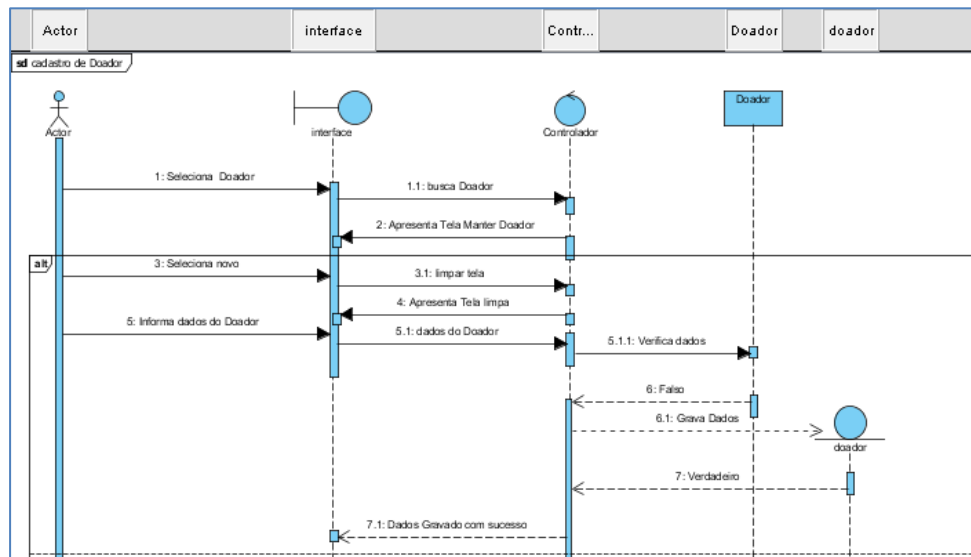


Figura 8 – Diagrama de Sequência Manter Doador
 Fonte: Autoria Própria

5.5.7 Diagrama de Sequência Cadastrar Doação

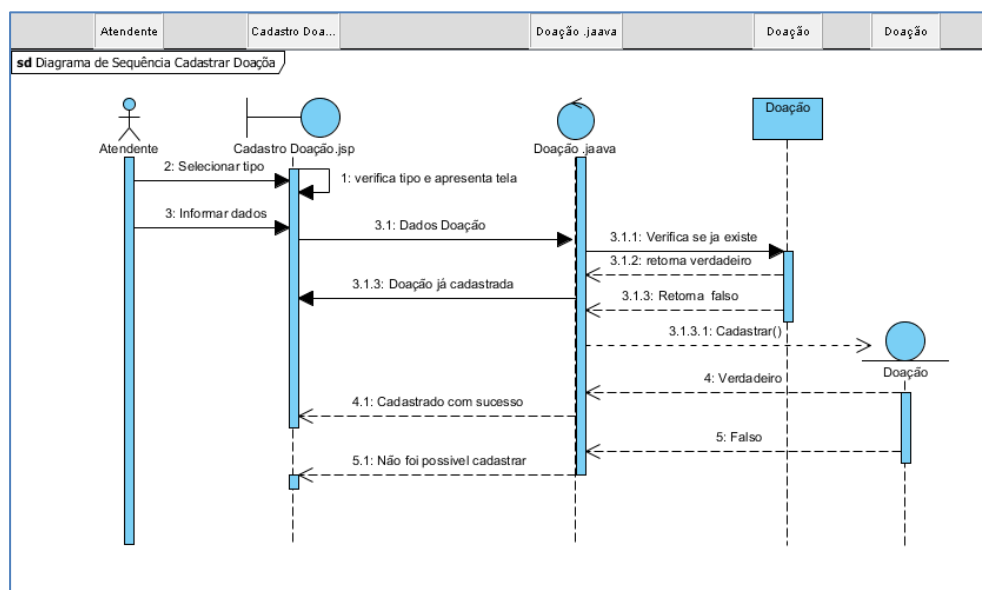


Figura 9 – Diagrama de Sequência Cadastrar Doação
 Fonte: Autoria Própria

5.6 Diagrama de Entidade de Relacionamento (DER)

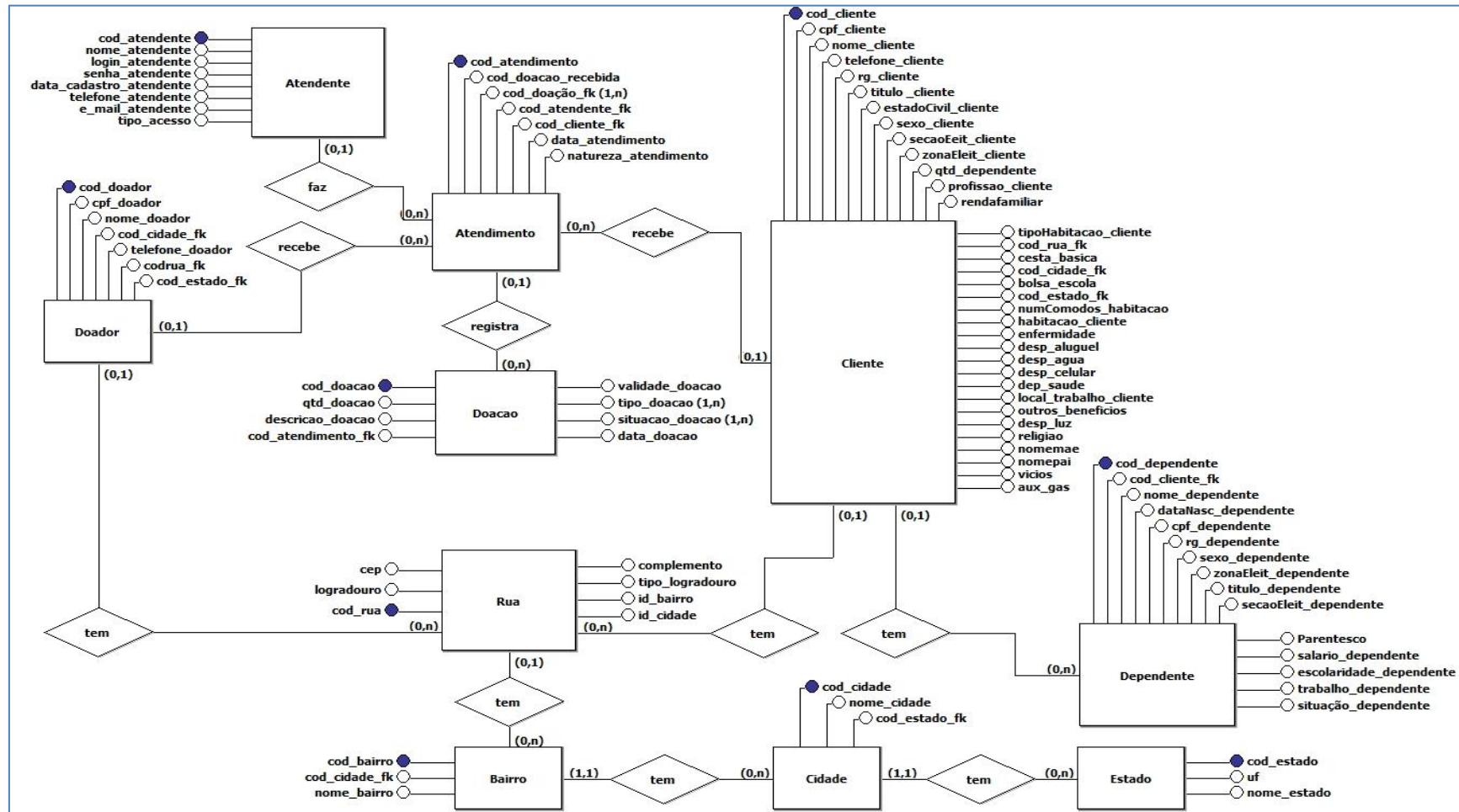


Figura 10 – Diagrama de Entidade de Relacionamento (DER)

Fonte: Autoria Própria

5.7 Diagrama de Classes

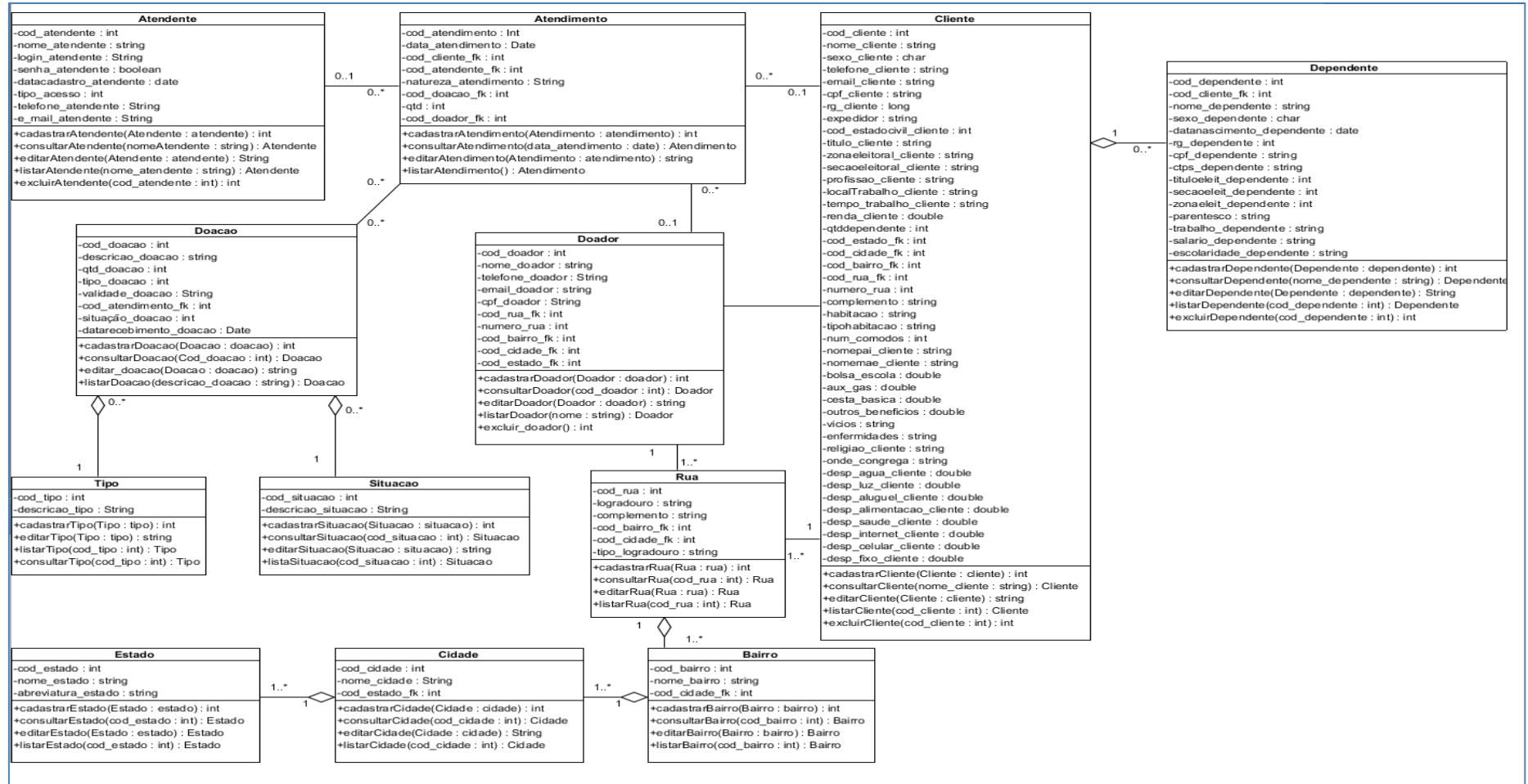


Figura 11 – Diagrama de Classe
 Fonte: Autoria Própria

5.8 Diagrama de Classes(MVC)

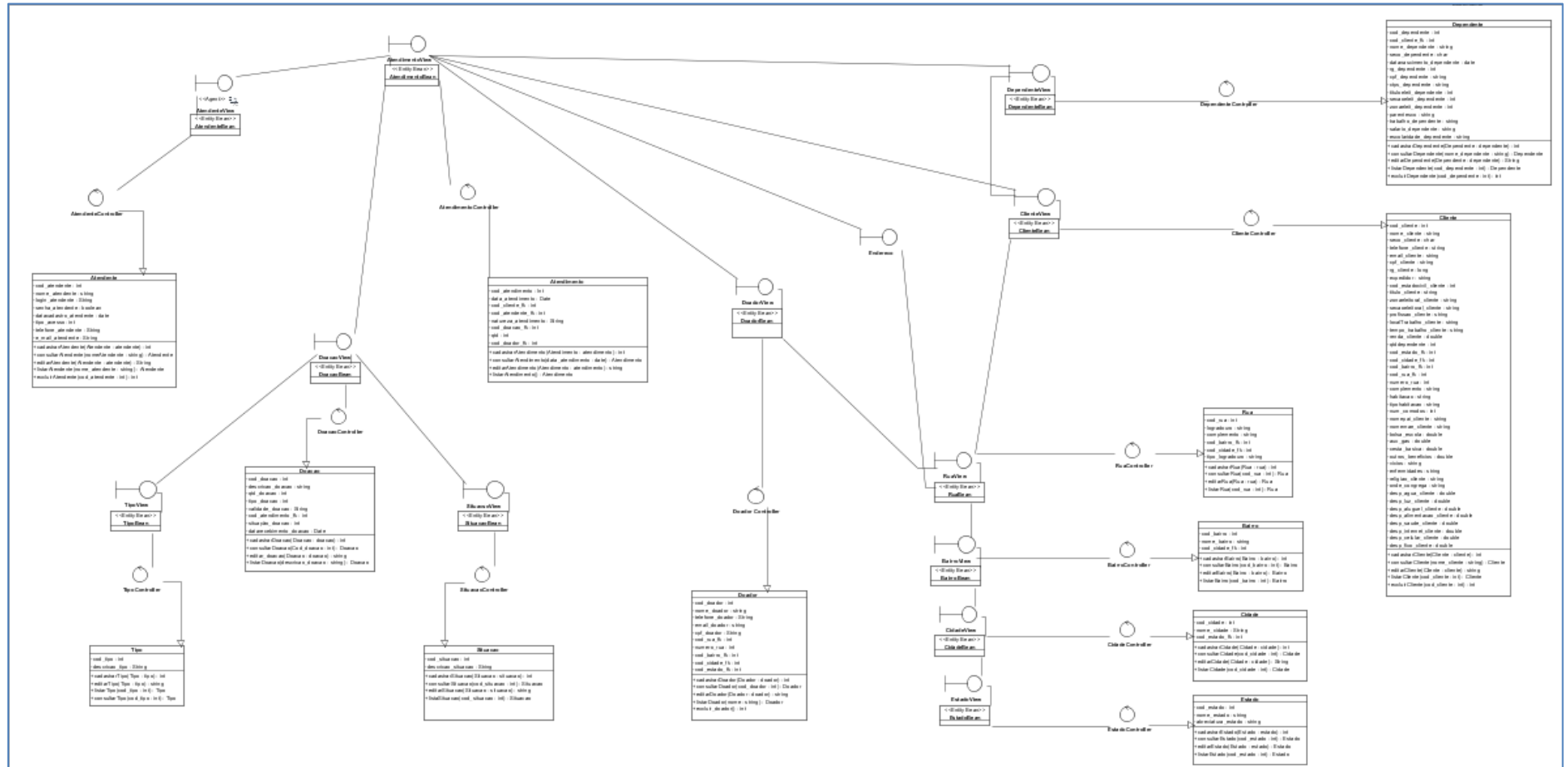


Figura 12 – Diagrama de Classe (MVC)
Fonte: Autoria Própria

5.8.1 Diagrama Classes Atendente e Atendimento (MVC)

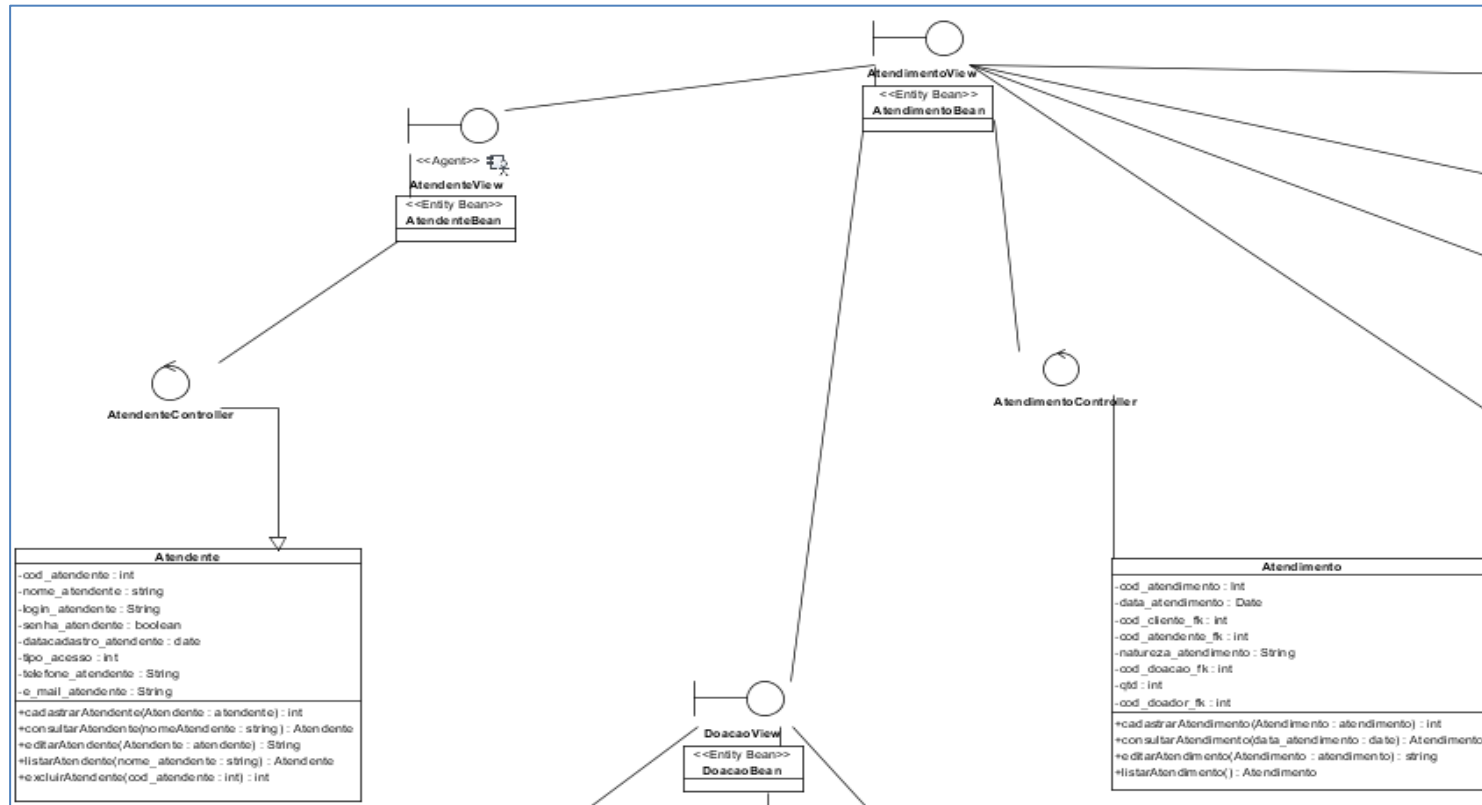


Figura 13 – Diagrama de Classe Atendente e Atendimento
 Fonte: Autoria Própria

5.8.2 Diagrama Classes Endereço (MVC)

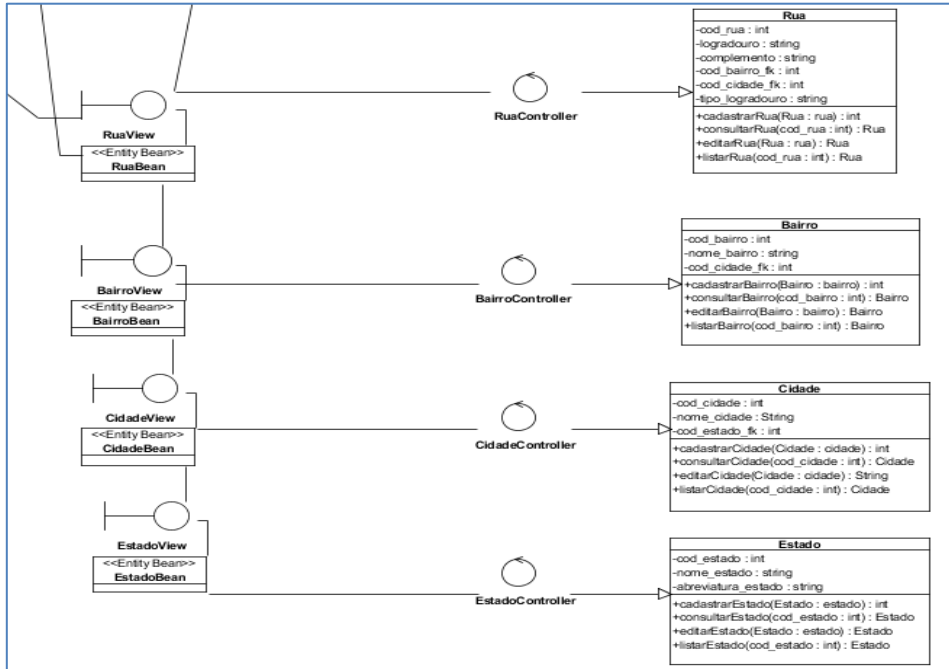


Figura 14 – Diagrama de Classe Endereço
Fonte: Autoria Própria

5.8.3 Diagrama Classes Doação (MVC)

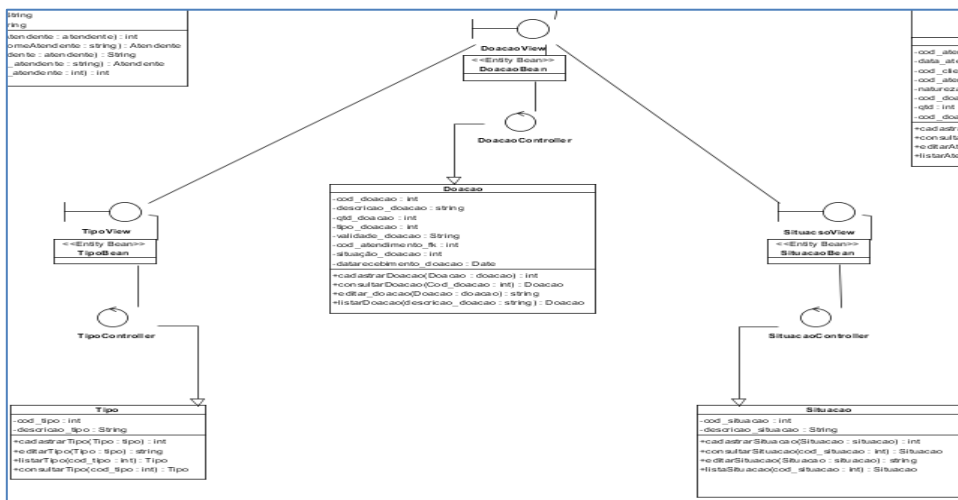


Figura 15 – Diagrama de Classe Doação
Fonte: Autoria Própria

5.8.4 Diagrama Classes Cliente e Dependente (MVC)

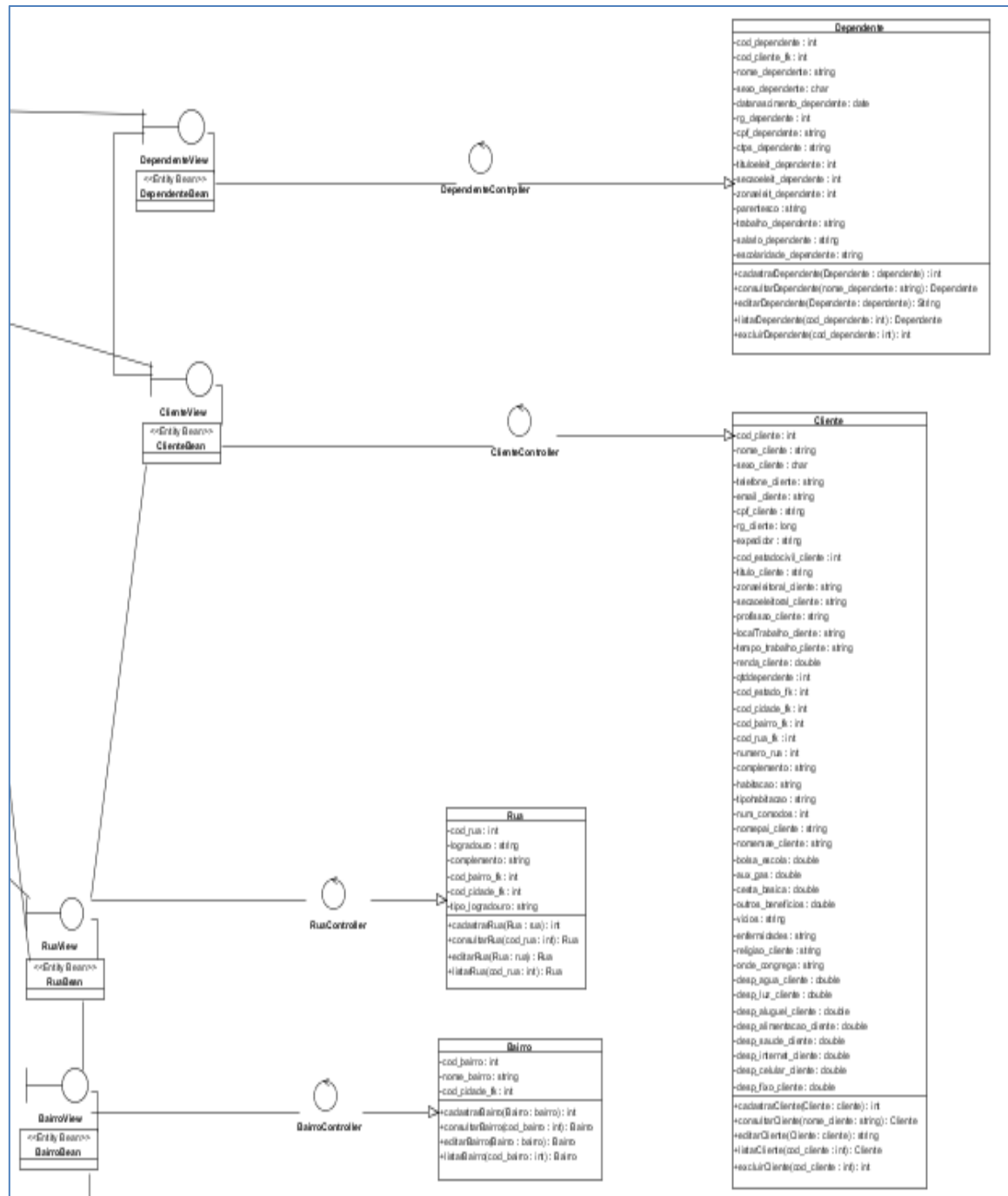


Figura 16 – Diagrama de Classe Cliente e Dependente
Fonte: Autoria Própria

Para efetuar a contagem de pontos por função foi primeiramente identificadas às funções que iriam ser incluídas e verificados os dados lógicos que seriam mantidos pela aplicação, logo após, realizada a contagem das funções dos dados os ALI (Arquivo Lógico Interno) e (AIE) Arquivo de Interface Externa, aplicando as regras de identificação, depois a contagem das funções de transações (IE) Input Externo e (OE) Output Externo, para finalizar foi realizado o fator de ajuste baseados em 14 características gerais do software.

5.10 Telas

As telas a seguir demonstram como foram utilizadas as tecnologias vista neste trabalho, após cada tela segue informações correspondentes e o código.

A tela abaixo é a de login utilizando o Bootstrap.



Figura 18 - Tela de Login
Fonte: Autoria Própria

Código de chamada do Bootstrap utilizado nas páginas web para visualização do programa em Tablet, Smartphone e Netbook, podendo ainda utilizar outros tamanhos personalizados.

```
<link rel="stylesheet" type="text/css" href="/resources/css/bootstrap-theme.css" />
<link rel="stylesheet" type="text/css" href="/resources/css/bootstrap.css" />
<link rel="stylesheet" type="text/css" href="/resources/css/bootstrap-theme.min.css"/>
<link rel="stylesheet" type="text/css" href="/resources/css/bootstrap.min.css"/>
```

Tela Lista de Atendente

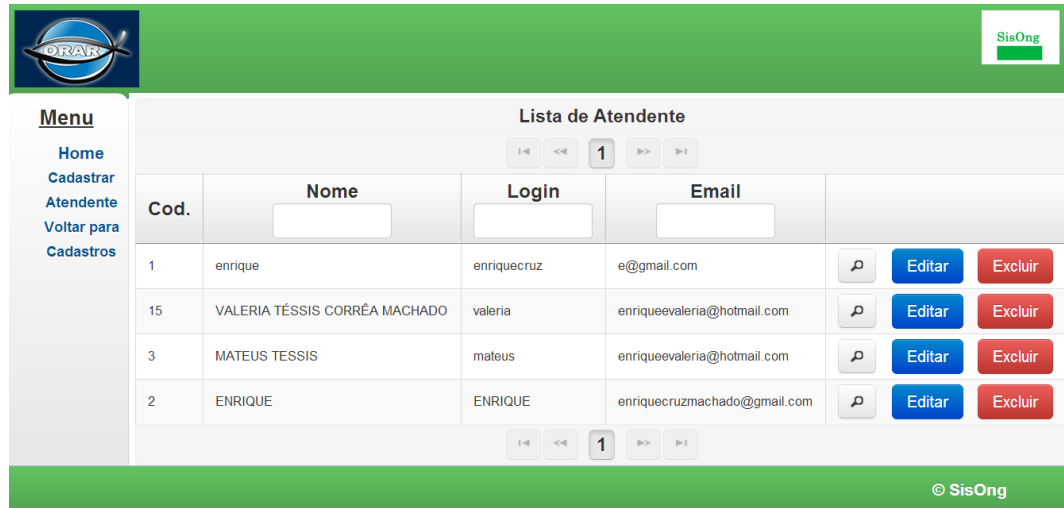


Figura 19 – Tela Lista Atendente
Fonte: Autoria Própria

O código abaixo mostra o xhtml com a utilização do Prime Faces da página acima todas as demais seguem o mesmo padrão.

```
<ui:composition xmlns:ui="http://java.sun.com/jsf/facelets" template=" ../template/templateF.xhtml"
  xmlns:h="http://xmlns.jcp.org/jsf/html"
  xmlns:f="http://xmlns.jcp.org/jsf/core"
  xmlns="http://www.w3.org/1999/xhtml"
  xmlns:p="http://primefaces.org/ui">
  <ui:define name="left">
    <li><a href="atendenteNovo.xhtml"><small>Cadastrar Atendente</small></a></li>
    <li><a href=" ../cadastros.xhtml"><small>Voltar para Cadastros</small></a></li>
  </ui:define>
  <ui:define name="content">
    <h:form>
      <p:dataTable value="#{atendenteBean.lista}" var="usu" paginator="true" rows="10"
        emptyMessage="Atendente não encontrado*">
        <f:facet name="header"> Lista de Atendente </f:facet>
        <p:column id="colunaCodigo" headerText="Cod." >
          <small> <h:outputText value="#{usu.codAtendente}"/> </small>
        </p:column>
        <p:column id="colunaCliente" headerText="Nome" filterBy="#{usu.nomeAtendente}">
          <small> <h:outputText value="#{usu.nomeAtendente}"/> </small>
        </p:column>
      </p:dataTable>
    </h:form>
  </ui:define>
</ui:composition>
```

Chamada da biblioteca PrimeFaces

Utilizando o Componente PrimeFaces

Tela de Informações

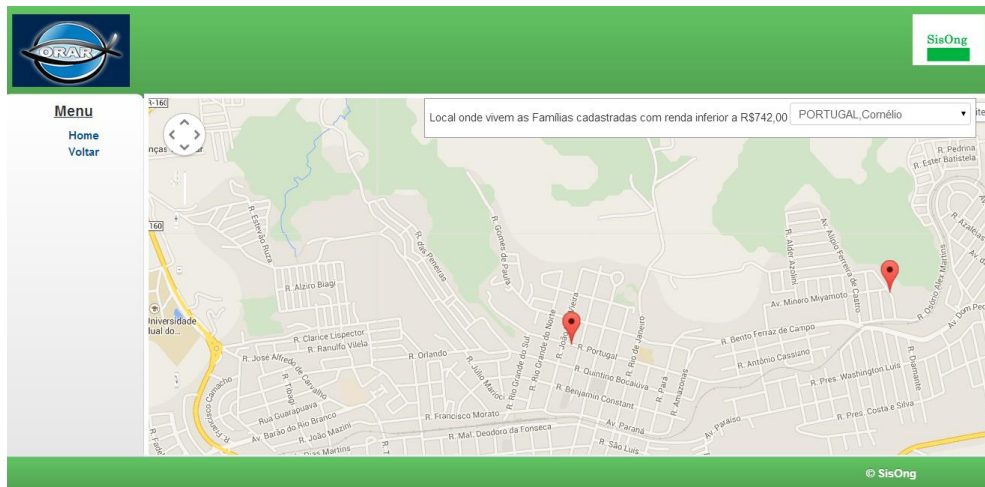


Figura 20 – Tela de Informações
Fonte: Autoria Própria

O código fonte abaixo mostra a utilização da API do Google Maps na aplicação para mapear os locais onde vivem as pessoas que receber doações e apresentar algumas informações sobre o local.

```

<script src="https://maps.googleapis.com/maps/api/js?v=3.exp"></script>
<div id="map-canvas"></div>
<script>
  var geocoder;
  var map;

  function initialize() {
    geocoder = new google.maps.Geocoder();
    var latlng = new google.maps.LatLng(-23.176815, -50.643785);
    var mapOptions = {
      zoom: 15,
      center: latlng
    };
    map = new google.maps.Map(document.getElementById('map-canvas'), mapOptions);
  }

  function codeAddress() {
    var address = document.getElementById('address').value;
    geocoder.geocode({'address': address}, function(results, status) {
      if (status === google.maps.GeocoderStatus.OK) {
        map.setCenter(results[0].geometry.location);
        var marker = new google.maps.Marker({
          map: map,
          position: results[0].geometry.location
        });
      }
    });
  }

```

Referência à biblioteca da API do Google Maps.

Atribuindo as coordenadas e o zoom desejado ao abrir a página.

Função que recebe o nome da rua, procura no mapa e adiciona o marcador.

```

    } else {
        alert('Geocode was not successful for the following reason: ' + status);
    }
    });
}

```

```


google.maps.event.addDomListener(window, 'load', initialize);

```

```

</script>
</h:form>
<div id="panel">
    Local onde vivem as Famílias cadastradas com renda inferior a R$742,00
    <h:selectOneMenu id="address" onchange="codeAddress()">
        <f:selectItem itemValue="" itemLabel=" Rua :"/>
        <f:selectItems value="#{clienteBean.getListarenda(742)}" var="rua"
itemValue="#{rua.codRuaFk.logradouro},Cornélio"
itemLabel="#{rua.codRuaFk.logradouro},Cornélio"/>
    </h:selectOneMenu>
</div>

```



Busca no banco todos os endereços de clientes com renda inferior ao valor estipulado, conforme for selecionado envia para a função que irá marcar no mapa o local.

6. PRINCIPAIS DIFICULDADES

No decorrer do projeto as principais dificuldades encontradas foram relacionadas às mudanças constante nos requisitos do sistema, motivo este, que fez com que houvesse a necessidade de ser feita uma boa definição da priorização dos requisitos e a alteração do cronograma para que o programa pudesse ser desenvolvido em tempo hábil.

Outra dificuldade foi trabalhar com Java Server Faces e outras tecnologia como Prime Faces e Bootstrap em conjunto.

7. CONSIDERAÇÕES FINAIS

Com o desenvolvimento do Software proposto e descrito no presente trabalho, a ONG ORAR poderá obter do fluxo de suas informações registradas, um controle maior de suas atividades e poderá melhorar seus serviços para com a comunidade carente. Este trabalho trará maior conhecimento de como as famílias estão sendo atendidas e quantas estão sendo beneficiadas por esta entidade, também proporcionara a apresentação junto aos órgãos competentes dos trabalhos realizados através do gerenciamento do fluxo dessas informações.

Futuramente serão desenvolvidos os demais requisitos que apareceram durante o desenvolvimento do Software e que ficaram fora do cronograma inicial devido a análise e a priorização dos requisitos que foram baseados nas necessidades específicas desta ONG, porém, devido ao tempo não ser suficiente para atender a todos, eles foram suprimidos em parte para que o projeto pudesse ser desenvolvido em tempo hábil.

O resultado obtido com o trabalho foi altamente satisfatório, pois foi possível aprender tecnologias novas e exercitar as matérias expostas nas disciplinas cursadas, além de auxiliar a entidade em suas necessidades.

REFERÊNCIAS

CIVICI, Çağatay. **Prime Face** User guide 4.0, Prime Faces Documentation. Disponível em: <<http://www.primefaces.org/documentation.html>> Último acesso em Outubro de 2014.

Cybis , Walter. **ErgoList**, Disponível em:<<http://www.labiutil.inf.ufsc.br/ergolist/index.html>>. Último acesso em Novembro de 2014.

EasyBus, **Ong Facil**. Disponível em: < <http://portalongfacil.com.br/>>/ Último acesso em novembro 2014.

GAMMA, E. et al. **Padrões de projeto**. [S.l.]: São Paulo: Bookman, 2006.

Gemini , **Mastermanager**. Disponível em: <<http://www.mastermanager.com.br/>>. Último acesso em outubro 2014.

Google Maps, 2014. **API Javascript do Google Maps v3**. Disponível em: <<HTTPS://developers.google.com/maps/documentation/javascript/tutorial>> Último acesso em janeiro de 2014.

KNIBERG, Henrik; SUTHERLAND, Jeff, 2007, InfoQ.com. **Scrum e XP direto das Trincheiras como fazemos Scrum**.

Otto, Mark, 2014. **Bootstrap**. Disponível em: <<http://getbootstrap.com/2.3.2/index.html>> Último acesso em Outubro de 2014.

Oracle, 2013. **NetBeans 7.4**. Disponível em : <<https://netbeans.org/community/releases/74/>> Último acesso em Setembro de 2014.

Oracle, 2014. **Criando o futuro com Java**. Disponível em: <<https://www.oracle.com/java/index.html>>. Último acesso em novembro de 2014.

PRESSMAN, Roger S. São Paulo, 1995, Makron Books, **Engenharia de software**.

PRESSMAN, Roger. S. São Paulo: McGraw-Hill, 2006. **Engenharia de Software** 6a edição. [S.l.]:

PRESSMAN, Roger S. Técnicas de Teste de Software. In: Engenharia de software. 6.ed. Rio de Janeiro: McGraw-Hill, 2006.p.327-332.

Pereira Neto, Álvaro, Editora Erika, 2003. **PostgresSQL**.

Salgado, Luiz A. Meirelles. São Paulo, 2000, Bookman, **Padrões de Projeto**.

SEMAD,2014 **Organização não governamental**. Disponível em: <<http://www.meioambiente.mg.gov.br/organizacoes-nao-governamentais>> Acessado em outubro 2014.

Terceiro Setor no Brasil, Disponível em: <<http://www.terceirosetoronline.com.br/terceiro-setor-no-brasil/>> Último acesso em dezembro de 2013.

Valdés, Damián Pérez, 2007. **O que é Java Script?**. Disponível em: <<http://www.maestrosdelweb.com/que-es-javascript/>> Último acesso em Dezembro de 2014.

Wikipédia, **Organização não governamental**. Disponível em: <http://pt.wikipedia.org/wiki/Organiza%C3%A7%C3%A3o_n%C3%A3o_governamental > Último acesso em outubro 2014.

Wikipédia, **Engenharia de Software**. Disponível em: <http://pt.wikipedia.org/wiki/Engenharia_de_software> Último acesso em outubro 2014.