



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

LEONARDO RODRIGUES BASTELLI

**IMPLEMENTAÇÃO DE WEBAPP COM PRIMEFACES MOBILE PARA
GESTÃO DE ORDENS DE SERVIÇOS E VISITAS**

Assis
2013

LEONARDO RODRIGUES BASTELLI

**IMPLEMENTAÇÃO DE WEBAPP COM PRIMEFACES MOBILE PARA
GESTÃO DE ORDENS DE SERVIÇOS E VISITAS**

Trabalho de Conclusão de Curso apresentado ao
Instituto Municipal de Ensino Superior de Assis,
como requisito do Curso de Graduação

Orientador: Prof. Esp. Guilherme de Cleve Farto

Área de Concentração: Informática

Assis
2013

FICHA CATALOGRÁFICA

BASTELLI, Leonardo Rodrigues

Implementação de webapp com *primefaces mobile* para gestão de ordens de serviços e visitas / Leonardo Rodrigues Bastelli. Fundação Educacional do Município de Assis – FEMA – Assis, 2013.

85p.

Orientador: Prof. Esp. Guilherme de Cleve Farto

Trabalho de Conclusão de Curso – Instituto Municipal de Ensino Superior de Assis – IMESA

1. PrimeFaces Mobile 2. Java 3. GOSV

CDD: 001.61
Biblioteca da FEMA

IMPLEMENTAÇÃO DE WEBAPP COM PRIMEFACES MOBILE PARA GESTÃO DE ORDENS DE SERVIÇOS E VISITAS

LEONARDO RODRIGUES BASTELLI

Trabalho de Conclusão de Curso apresentado ao
Instituto Municipal de Ensino Superior de Assis,
como requisito do Curso de Graduação, analisado
pela seguinte comissão examinadora:

Orientador: Prof. Esp. Guilherme de Cleva Farto

Analisador (1): Prof. Dr. Alex Sandro Romeo de Souza Poletto

Assis
2013

DEDICATÓRIA

Dedico este trabalho primeiramente a Deus que me deu forças para concluir o objetivo proposto. Dedico também à minha família que com muita paciência e amor me deu suporte sem questionar minha ausência.

AGRADECIMENTOS

Primeiramente a Deus, que me deu força para concluir esse projeto, a minha namorada Daniele, meu filho Victor, pois sem eles nada disso seria possível, a minha mãe Leiliana e minha querida avó Dejanira que me deram apoio incondicional em todos os momentos.

Ao meu orientador Esp. Guilherme de Cleve Farto, que me deu tranquilidade durante todo o desenvolvimento de meu trabalho de conclusão de curso.

Aos colegas de classe que conquistei ao longo desses três anos de aprendizado.

Aos amigos de todas as horas Jônatas e Gabriel, pelos momentos de descontração proporcionados durante esses três anos, pelo incentivo e colaboração para elaboração e término desse trabalho.

E a todos que colaboraram direta ou indiretamente na execução deste trabalho.

“O único homem que está isento de erros é aquele que não arrisca acertar [...].”

Albert Einstein (1879-1955)

RESUMO

O presente trabalho descreve o desenvolvimento de um sistema de gestão de ordem de serviços e visitas, com ênfase nos módulos de agendamento de visitas e emissão de ordem de serviço. O objetivo do sistema é a diminuição do volume de papéis e tempo gasto na geração do serviço manualmente. O projeto foi desenvolvido utilizando a linguagem *Java* com a plataforma *Java EE*, em conjunto dos frameworks, *JPA* para mapeamento objeto e relacional, *PrimeFaces Mobile*, para o desenvolvimento da parte visual e *JSF* que disponibiliza um conjunto de componentes específicos para o desenvolvimento rápido.

Palavras-chave: Desenvolvimento de aplicação *Web Java EE*; *Java*; *JSF*; *PrimeFaces Mobile*; *MySQL*.

ABSTRACT

The current work describes the development of a work order and visiting management system, with emphasis on the scheduling visits and work order issue modules. The system goal is to diminish the amount of paper and the time spent in the manually generation of the service. The project was developed using the Java language with the Java EE platform, along with the frameworks JPA to promote the object and relational mapping, PrimeFaces Mobile to develop the visual interface, and the JSF which presents a set of specific components aimed to the agile development.

Key-words: Java EE Web Application Development; Java; JSF; PrimeFaces Mobile; MySQL.

LISTA DE ILUSTRAÇÕES

Figura 1 – Compilação de código C	24
Figura 2 – Compilação de aplicação JAVA	24
Figura 3 – Módulos PrimeFaces.....	29
Figura 4 – Mapa Mental	31
Figura 5 – Caso de Uso Administrador	33
Figura 6 – Caso de Uso Comum Administrador e Usuário.....	34
Figura 7 – Caso de Uso Comum Administrador e Usuário.....	35
Figura 8 – UC1 Diagrama de caso de uso Efetuar Login	35
Figura 9 – UC2 Diagrama de caso de uso Manter Cliente	35
Figura 10 – UC3 Diagrama de caso de uso Manter Empresa.....	37
Figura 11 – UC4 Diagrama de caso de uso Manter Cidades	38
Figura 12 – UC5 Diagrama de caso de uso Manter Estados	38
Figura 13 – UC6 Diagrama de caso de uso Manter Serviços	39
Figura 14 – UC7 Diagrama de caso de uso Manter Visitas.....	40
Figura 15 – UC8 Diagrama de caso de uso Manter Setor.....	41
Figura 16 – UC9 Diagrama de caso de uso Manter Ordem de Serviço	42
Figura 17 – UC10 Diagrama de caso de uso Manter Funcionário.....	42
Figura 18 – UC11 Diagrama de caso de uso Manter Localização Serviço	43
Figura 19 – UC12 Diagrama de caso de uso Manter Localização Visita.....	44
Figura 20 – UC13 Diagrama de caso de uso Manter Status Serviço	45
Figura 21 – UC14 Diagrama de caso de uso Manter Status Visita	45
Figura 22 – UC15 Diagrama de caso de uso Manter Tipo Serviço	46
Figura 23 – UC16 Diagrama de caso de uso Movimentar Agendamento de Visitas .	47
Figura 24 – UC17 Diagrama de caso de uso Movimentar Equipe.....	48
Figura 25 – UC18 Diagrama de caso de uso Movimentar Geolocalização	49
Figura 26 – UC19 Diagrama de caso de uso Movimentar Ordem de Serviço	50
Figura 27 – UC20 Diagrama de caso de uso Movimentar Status Serviço.....	51
Figura 28 – UC21 Diagrama de caso de uso Movimentar Status Visitas	52
Figura 29 – UC22 Diagrama de caso de uso Emitir Relatório de Agendamentos	53
Figura 30 – UC23 Diagrama de caso de uso Emitir Relatório de Clientes	53
Figura 31 – UC24 Diagrama de caso de uso Emitir Relatório de Serviços	54

Figura 32 – UC25 Diagrama de caso de uso Emitir Relatório de Setores.....	55
Figura 33 – UC26 Diagrama de caso de uso Emitir Relatório de Visitas	56
Figura 34 – UC27 Diagrama de caso de uso Emitir Relatório de Funcionários	57
Figura 35 – Diagrama de Atividade Cadastrar Ordem de Serviço.....	58
Figura 36 – Diagrama de Atividade Cadastrar Visita.....	59
Figura 37 – Diagrama de Atividade Cadastrar Agendamento	60
Figura 38 – Diagrama de Sequencia Cadastrar Ordem de Serviço	61
Figura 39 – Diagrama de Sequencia Cadastrar Visita	62
Figura 40 – Diagrama de Sequencia Cadastrar Agendamento	63
Figura 41 – Diagrama de Classe.....	64
Figura 42 – Diagrama Entidade Relacionamento.....	64
Figura 43 – Estrutura Analítica do Projeto.....	66
Figura 44 – Sequenciamento das Atividades	67
Figura 45 – Divisão de Camadas	69
Figura 46 – Classe Converter Cliente	70
Figura 47 – Classe Usuário Dao	71
Figura 48 – Classe Usuário Model	72
Figura 49 – Classe JPAUtil.....	73
Figura 50 – Classe ClienteMB.....	74
Figura 51 – Página <i>JSF</i> consultaOrdemServico.xhtml	75
Figura 52 – Página Inicial sistema GOSV	76
Figura 53 – Página Menu sistema GOSV	76
Figura 54 – Página Cadastro de cliente sistema GOSV.....	77
Figura 55 – Página Cadastro de cliente (Mobile) sistema GOSV.....	78
Figura 56 – Página Consulta cliente sistema GOSV	79
Figura 57 – Página Consulta cliente (Mobile) sistema GOSV	80

LISTA DE TABELAS

Tabela 1 – Efetuar Login	35
Tabela 2 – Manter Cliente	35
Tabela 3 – Manter Empresa	36
Tabela 4 – Manter Cidades	37
Tabela 5 – Manter Estados	38
Tabela 6 – Manter Serviços.....	38
Tabela 7 – Manter Visitas.....	39
Tabela 8 – Manter Setor.....	40
Tabela 9 – Manter Ordem de Serviço	41
Tabela 10 – Manter Funcionário.....	42
Tabela 11 – Manter Localização Serviço	42
Tabela 12 – Manter Localização Visita.....	43
Tabela 13 – Manter Status Serviço	44
Tabela 14 – Manter Status Visitas.....	45
Tabela 15 – Manter Tipo Serviço	45
Tabela 16 – Movimentar Agendamento de Visitas	46
Tabela 17 – Movimentar Equipe.....	47
Tabela 18 – Movimentar Geolocalização	48
Tabela 19 – Movimentar Ordem de Serviço.....	49
Tabela 20 – Movimentar Status Serviços	50
Tabela 21 – Movimentar Status Visitas	51
Tabela 22 – Emitir Relatório de Agendamentos	52
Tabela 23 – Emitir Relatório de Clientes	53
Tabela 24 – Emitir Relatório de Serviços	53
Tabela 25 – Emitir Relatório de Setores.....	54
Tabela 26 – Emitir Relatório de Visitas	55
Tabela 27 – Emitir Relatório de Funcionário	56
Tabela 28 – Orçamento Analista / Programador	66
Tabela 29 – Orçamento Valor Total	67

SUMÁRIO

1 – INTRODUÇÃO	15
1.1 – OBJETIVOS	17
1.2 – JUSTIFICATIVAS	18
1.3 – PÚBLICO ALVO	19
1.4 – ESTRUTURA DO TRABALHO	19
2 – TECNOLOGIAS E FERRAMENTAS DE DESENVOLVIMENTO	21
2.1 – JAVA	21
2.1.1 – Principais características Java	22
2.1.2 – Máquina Virtual Java	23
2.2 – PLATAFORMA JAVA ENTERPRISE EDITION (JEE)	25
2.3 – JAVASERVER FACES (JSF)	26
2.4 – JAVA PERSISTENCE API (JPA)	27
2.5 – BANCO DE DADOS MySQL	28
2.6 – FRAMEWORK PRIMEFACES	29
2.6.1 – PrimeFaces Mobile	30
3 – ANÁLISE E ESPECIFICAÇÃO DO SISTEMA	31
3.1 – MAPA MENTAL	31
3.2 – LISTA DE REQUISITOS	32
3.3 – DIAGRAMA E ESPECIFICAÇÃO DE CASOS DE USO	33
3.4 – DIAGRAMA DE ATIVIDADES	57
3.5 – DIAGRAMA DE SEQUENCIA	60
3.6 – DIAGRAMA DE CLASSE	62
3.7 – MODELO ENTIDADE-RELACIONAMENTO	63
4 – ESTRUTURA DO PROJETO	65
4.1 – ESTRUTURA ANALÍTICA DE TRABALHO	65
4.2 – SEQUENCIAMENTO DAS ATIVIDADES	66
4.3 – ORÇAMENTO	66
5 – IMPLEMENTAÇÃO DO PROJETO	69
5.1 – ORGANIZAÇÃO ESTRUTURAL DO PROJETO JAVA	70
5.2 – INTERFACES DA APLICAÇÃO	76

6 – CONCLUSÃO	81
6.1 – TRABALHOS FUTUROS	82
REFERÊNCIAS	83

1 – INTRODUÇÃO

Atualmente, vivemos na era da informação e mobilidade, exigindo das organizações estratégias eficientes em seus processos a fim de torná-los mais ágeis e fáceis por meio da utilização de ferramentas inteligentes fornecidas por sistemas de informação (BAZZOTTI; GARCIA, 2007).

A falta de sistemas de informação em um ambiente corporativo agrega um déficit em seus processos, devido à emissão de ordens de serviços em fichas de papéis, além do gasto e desordem no armazenamento de documentos e da necessidade de um funcionário específico para essa função.

Na prestação de serviço, principalmente em relação a equipamentos eletrônicos, deve-se ter controle com relação à entrada, discriminação completa e detalhada sobre o aparelho, relatório de danos e peças substituídas. Dessa forma, eleva-se o número de fichas já que, para cada equipamento, um documento contendo as informações mencionadas anteriormente deve ser elaborado.

Além do elevado número de fichas, uma empresa não informatizada corre risco de perda tanto em papéis quanto em serviços não executados no prazo estabelecido. O agendamento de visitas é desgastante ao funcionário devido à demora no preenchimento de papéis, pois são muitos dados importantes como endereço, telefone e contato eletrônico. Além do preenchimento de fichas, torna-se indispensável o uso de uma agenda para lembretes sobre horário, data e informações essenciais sobre a visita e serviços prestados.

Uma aplicação *Web* é um sistema acessado por uma rede, como a Internet, por meio de um navegador de páginas. Atualmente, a Internet oferece uma infinidade de aplicações com conteúdo dinâmico e personalizado, devido a tecnologias que possibilitaram a evolução de páginas antes estáticas. Dentre as tecnologias disponíveis para o desenvolvimento de aplicações *Web*, destacam-se *Servlets* e páginas *Java Server Pages (JSP)* (TEMPLE et al., 2004).

Os *Servlets* são classes *Java* desenvolvida para que, quando publicadas junto a um servidor de aplicação que implemente um *Servlet Container*, sejam capazes de

receber, manipular e responder a requisições recebidas de clientes (GONÇALVES, 2008).

O *JavaServer Faces (JSF)* é uma tecnologia padrão da plataforma *Java Enterprise Edition (JEE)* utilizada para criar aplicações *Web* por meio de um conjunto de componentes específicos para o desenvolvimento rápido de aplicações ou *Rapid Application Development (RAD)* e é altamente baseada nas tecnologias de *Servlets* e *JSP* (GOMES, 2008).

Uma das principais vantagens em se desenvolver uma aplicação *Web* com a plataforma *Java* é a facilidade de utilização de *Model-View-Controller (MVC)*, um padrão estrutural que oferece benefícios significantes e que objetiva separar a solução em partes ou camadas, de modo que as regras de negócio permaneçam isoladas das camadas de visualização e controle. Por meio do uso de MVC, a aplicação torna-se mais simples quanto ao quesito de manutenibilidade além de melhorar a organização do projeto (GONÇALVES, 2008).

Além de soluções *Web*, um novo tipo de aplicação, chamada de *WebApp*, tem ganhado espaço no mercado corporativo, fazendo-se cada vez mais presentes nas estratégias de negócios de grandes e pequenas empresas. Uma *WebApp* é um sistema ou aplicação baseada na *Web* capaz de ser acessado por navegadores de Internet, como *Internet Explorer* e *Google Chrome*, assim como por dispositivos móveis como *smartphones* e *tablets* sem ser necessário uma refatoração do projeto *Web* já implementado (LOCKOW; MELO, 2010) e (WAGNER; MÄHLMANN, 2012).

Com o uso do *framework PrimeFaces Mobile* é possível à implementação de *WebApps* baseadas em JSF de maneira rápida, simplificada e produtiva, permitindo que aplicações *Web* possam visualizadas em navegadores padrões assim como em dispositivos móveis. A tecnologia *PrimeFaces Mobile* faz uso de *jQuery Mobile*, uma plataforma que possibilita a execução de aplicações em diversos ambientes e dispositivos finais (WAGNER, 2012).

Pesquisas realizadas apontam o aumento significativo no uso de aparelhos portáteis, contribuindo para que empresas adotem tecnologias de gestão para dispositivos móveis ou *Mobile Device Management (MDM)*. A expectativa é de que, até 2017, 67% das empresas adotem a tecnologia MDM, devido à preocupação com

a segurança em transmissão de dados de *smartphones* e *tablets* (COMPUTERWORLD, 2013).

Devido às dificuldades existentes, a aplicação *WebApp* de Gestão de Ordens de Serviços e Visitas (GOSV) objetiva disponibilizar um produto capaz de melhorar os processos de controle de tarefas e atendimento aos clientes assim como elevar as oportunidades comerciais e técnicas para o mercado de atuação de uma determinada empresa.

O produto GOSV apresenta diversas funcionalidades implementadas com o intuito de facilitar a prestação de serviços e visitas a clientes de uma empresa que faz uso da aplicação, permitindo um melhor controle de diversas informações tais como tipos de serviços, clientes, funcionários, ordens de serviço e visitas além do armazenamento de históricos sobre os visitas e serviços realizados.

1.1 OBJETIVOS

A informatização de empresas tem-se tornado uma necessidade real, e não mais optativa, por motivos como o tempo gasto com atividades realizadas manualmente, o extenso volume de papéis e documentos gerados para simples tarefas empresariais e até mesmo pela grande falta de controle nos processos da organização.

A empresa VNGT Indústria e Comércio de Equipamentos Eletrônicos Ltda., localizada em Assis/SP e atuante no mercado de telecomunicações há dez anos, será a pioneira na utilização da *WebApp* GOSV. O setor de ordens de serviços e visitas ainda não é informatizado, apesar de ser um dos mais importantes devido à prestação de serviços de manutenção de equipamentos eletrônicos.

Atualmente, a organização e a execução de serviços são feitas por meio de papéis seguindo o fluxo de que a ordem de serviço deve ser emitida por equipamento e, após a emissão, a mesma é enviada ao técnico responsável por realizar o orçamento assim como elaborar a relação de peças a serem utilizadas. Após essa atividade técnica, a ficha retorna para o setor de manutenção objetivando formalizar

o orçamento junto ao cliente e, com a devida aprovação, a área técnica inicia o serviço solicitado. O agendamento de visitas também é realizado manualmente com o auxílio de uma agenda contendo diversas informações, entretanto é impossível a emissão de relatórios acerca de estatísticas, valores financeiros e históricos de serviços ou visitas.

Com o objetivo de minimizar e até mesmo eliminar diversas dificuldades inerentes a organizações ainda não informatizadas ou que fazem uso de *softwares* que não atendem às necessidades específicas, espera-se desenvolver uma nova solução computacional na área de Gestão de Ordens de Serviços e Visitas.

A partir do uso da aplicação proposta neste trabalho acadêmico, pretende-se elevar a eficiência em atendimentos a clientes, agilizar processos da organização, reduzir despesas por meio de tecnologias modernas e emergentes bem como ampliar a comunicação e relacionamento entre clientes e empresa e o modo de mapeamento de visitas que será realizado pelo mecanismo de geolocalização *Google Maps*.

1.2 JUSTIFICATIVAS

O acesso à Internet vem crescendo gradativamente tornando os sistemas *Web* um grande atrativo para as empresas devido à mobilidade de acesso em qualquer lugar. Além disso, é preciso considerar a redução de recursos de *hardware* e *software* dos computadores em que a aplicação será utilizada, já que esta é disponibilizada em um servidor *Web* responsável por todo o processamento a ser realizado.

Um levantamento divulgado pela Telebrasil mostra que o acesso à banda larga vem crescendo rápido devido ao uso de terminais móveis 3G. O acesso chegou a 88,3 milhões em Janeiro de 2013, tendo alcançado um aumento de 37% em relação ao mesmo período do ano passado. Em um ano, o número de acessos cresceu 47%, chegando a 67,6 milhões de conexões. Dessas, 79,73% eram de dispositivos 3G, incluindo *smartphones* e *tablets* (EXAME, 2013).

As empresas do mundo contemporâneo procuram por soluções seguras para seus setores de forma que a informação seja sigilosa, segura e que não ocorram perdas.

Espera-se que o *software* proposto contribua de forma efetiva para um melhor e mais detalhado controle sobre serviços e visitas a serem realizados pelos usuários, além de possibilitar a emissão de informações e relatórios, em tempo real e com mobilidade, de forma que as tarefas estejam mais bem organizadas e distribuídas e sejam efetuadas com qualidade e rapidez.

1.3 PÚBLICO ALVO

A aplicação a ser desenvolvida neste trabalho visa atender às necessidades de empresas de vários nichos de mercado que possuem a execução de ordens de serviços e visitas como uma das principais tarefas organizacionais.

Dessa forma, nota-se que o produto final deste projeto de conclusão de curso poderá ser oferecido como uma solução moderna para diversas empresas que mantêm funcionários em campo, ou seja, fora do ambiente da empresa, tais como vendedores, consultores, representantes, técnicos e quaisquer outros cargos no âmbito externo.

Como forma de melhorar a imagem, a qualidade e a agilidade da empresa, quanto ao cumprimento de atividades externas, optou-se pela arquitetura e desenvolvimento de uma aplicação baseada nos conceitos de *WebApp*, possibilitando seu uso em dispositivos leves e móveis como *smartphones* e *tablets*.

1.4 ESTRUTURA DO TRABALHO

Este trabalho está estruturado nas seguintes partes:

- **Capítulo 1 – Introdução**
- **Capítulo 2 – Tecnologia e Ferramentas de Desenvolvimento**
- **Capítulo 3 – Análise e Especificação do Sistema**
- **Capítulo 4 – Estrutura do Projeto**
- **Capítulo 5 – Implementação do Projeto**

- **Capítulo 6 – Conclusão**
- **Referências**

2 – TECNOLOGIAS E FERRAMENTAS DE DESENVOLVIMENTO

O intuito deste capítulo é apresentar as tecnologias e ferramentas empregadas no desenvolvimento do sistema GOSV. A tecnologia *Java* foi escolhida para o desenvolvimento do *software* por ser uma plataforma e tecnologia orientada a objetos de alto nível, enquanto que o ambiente integrado de desenvolvimento, do inglês *Integrated Development Environment (IDE)*, adotado será o *Eclipse* devido à possibilidade de integração com novos componentes e o fácil acoplamento de novas bibliotecas, *frameworks* e diversas *Application Programming Interface (API)*.

Dessa forma, utilizar-se-á a tecnologia *Java* para o desenvolvimento da aplicação *Web* fazendo-se uso do banco de dados *MySQL*, um dos mais utilizados por grandes empresas, para a persistência e manipulação de dados por meio do *framework* de persistência *Java Persistence API (JPA)*, responsável pelo mapeamento objeto-relacional entre as classes de entidades do sistema e as tabelas da base de dados.

Para a camada visual responsável pela interação com os usuários, será utilizada a tecnologia de *JavaServer Faces (JSF)* e *PrimeFaces*, o que torna possível a implementação de aplicações *Web* capazes de serem executadas em navegadores de Internet, como *Internet Explorer* e *Chrome*, assim como em dispositivos móveis como *smartphones* e *tablets*.

O servidor de aplicação *Apache Tomcat* é um contêiner *Java* e um servidor *Web* ao mesmo tempo. Suporta a execução das tecnologias *JavaServlet* e *JavaServer Pages (JSP)*. Embora o *Tomcat* seja robusto o suficiente para ser utilizado em ambiente de produção, ele também permite integração com servidores como *Apache HTTP* (LOCKOW; MELO, 2010).

2.1 – Java

Java é uma plataforma de desenvolvimento em constante evolução concebida em 1995 pela *Sun Microsystems*. Inicialmente, a linguagem desenvolvida em 1991 pela equipe liderada por *James Gosling* tinha como objetivo equipamentos eletrônicos, porém com requisitos fundamentais, como robustez e código pequeno. Isso levou a

equipe a pesquisar modelos de implementação em *Pascal*, pois seu criador já havia pesquisado o design de uma linguagem portátil que gerava um código intermediário destinado a uma máquina hipotética. Devido a isso, surge o nome de uma máquina virtual que tornava possível a execução de um código de máquina próprio e independente do sistema operacional. Em razão da experiência do pessoal da *Sun* em *UNIX* que foi desenvolvido com a linguagem C e C++, os engenheiros optaram por desenvolver uma linguagem orientada a objetos. Assim, surge o nome *Oak* dado por *James Gosling* devido a uma árvore de carvalho que ele avistava pela janela de seu escritório. Entretanto, já existia uma linguagem com este nome e por isso surge o nome *Java*, baseado em um tipo de café (JAVAFREE, 2011) e (CAELUM, 2013).

Em 1992 o projeto *Green* entrega seu primeiro produto batizado de “*7” (*Star seven*), porém não despertou interesse na *Sun* em produzi-lo. Nos anos de 1993 e 1994, a equipe tentou destinar o produto a outras empresas, porém sem sucesso e então em meados de 1994 percebe que, com o avanço da internet e com uma grande rede interativa se formando, deveriam adaptar a linguagem para internet sendo projetada inicialmente para se mover em redes com internet, assim tornando possível sua execução dentro de *Browsers* com o conteúdo instantâneo. A plataforma *Java* foi apresentada formalmente pela *Sun* em 1995 em uma conferência o que provocou um grande interesse na linguagem devido à explosão da *Web* (HORSTMANN; CORNEL, 2010), (JAVAFREE, 2011) e (CAELUM, 2013).

2.1.1 – Principais características Java

A Plataforma *Java* possui muitos pontos-chave desde sua criação. Alguns deles serão apresentados nos tópicos seguintes:

- **SIMPLES:** a sintaxe da linguagem *Java* e uma versão simplificada da sintaxe do C++, não tendo a necessidade de arquivos de cabeçalho, aritmética de ponteiro, estruturas, uniões e sobrecarga de operadores.
- **ORIENTADA A OBJETOS:** recursos comparáveis aos do C++, com a diferença de que a herança múltipla foi substituída no *Java* pelo conceito mais simples de interfaces e o modelo de metaclasses.

- **DISTRIBUÍDA:** o *Java* dispõe de uma extensa biblioteca de rotinas para lidar com protocolos *TCP/IP* como *HTTP* e *FTP* de fácil uso através de *Sockets*.
- **SEGURA:** desde que a Plataforma foi concebida, seus criadores deram ênfase em tipos de segurança como estouro de pilha em tempo de execução, corrupção de memória fora do seu espaço de processo e leitura ou gravação de arquivos sem permissão.
- **ARQUITETURA NEUTRA:** formato de arquivo de objetos gerado pelo compilador é neutro em relação à arquitetura, o código compilado gera um *bytecode* interpretado pela (*JVM*) independente do sistema operacional.
- **INTERPRETADO:** interpretador *Java* executa *bytecode* em qualquer máquina em que o interpretador esteja instalado, os *bytecodes* são convertidos em código de máquina pelo compilador *just-in-time*.
- **MÚLTIPLOS THREADS:** maior capacidade de resposta interativa e melhor comportamento em tempo real de execução, pois praticamente todos os sistemas operacionais no mercado são multitarefas.
- **COLETOR DE LIXO:** em *Java* não há necessidade de se atentar ao gerenciamento de memória, pois o coletor de lixo se encarrega de mapear a memória de tempos em tempos verificando e liberando blocos não utilizados (HORSTMANN; CORNEL, 2010) e (DEITEL; DEITEL, 2010).

2.1.2 – Máquina Virtual Java

Um compilador *Java* gera um executável com instruções genéricas que são traduzidos pela (*JVM*) e repassados para o sistema operacional, instruções essas que são os *bytecodes* contidos em arquivos *.class*. A (*JVM*) é uma aplicação que simula um computador com seu próprio código de máquina (*assembly*), abstraindo tanto a camada de hardware como a de comunicação com o sistema operacional, em uma aplicação tradicional como o C a compilação do código gera um executável específico para o sistema operacional logo a tornando não portátil (SILVEIRA;

SILVEIRA; LOPES, MOREIRA; STEPPAT; KUNG, 2012), (DEITEL; DEITEL, 2010) e (CAELUM, 2013).

A Figura 1 ilustra a compilação de uma aplicação em C:

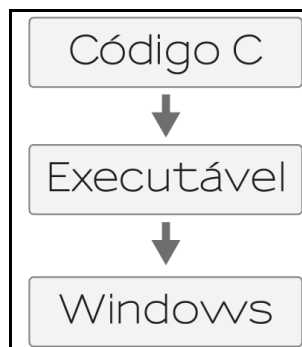


Figura 1 Compilação de código C. Fonte: (SILVEIRA; SILVEIRA; LOPES, MOREIRA; STEPPAT; KUNG, 2012).

Em uma aplicação Java o processo é parecido, porém o código é traduzido para *bytecodes* para que a (*JVM*) possa executar. Devido a essa métrica foi possível alcançar o desempenho e a portabilidade que a plataforma *Java* oferece (SILVEIRA; SILVEIRA; LOPES, MOREIRA; STEPPAT; KUNG, 2012) e (DEITEL; DEITEL, 2010).

A Figura 2 ilustra o processo que compilação de uma aplicação Java:

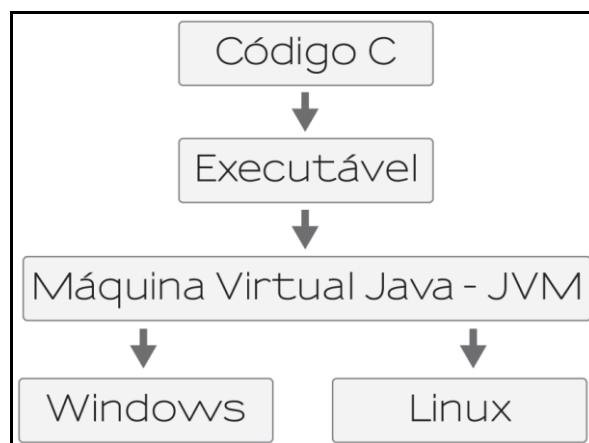


Figura 2 Compilação de aplicação Java. Fonte: (SILVEIRA; SILVEIRA; LOPES, MOREIRA; STEPPAT; KUNG, 2012).

A execução de uma aplicação *Java* foi aperfeiçoada com o passar do tempo. Em sua primeira versão, o *bytecode* era interpretado um por vez, na (*JVM*) atual é usado

o sistema de compilação *Just in Time (JIT)*. Essa técnica cria *Hot-Spots*, áreas de código executadas com maior frequência (DEV MEDIA 2013).

2.2 – PLATAFORMA JAVA ENTERPRISE EDITION (JEE)

A plataforma *Java Enterprise Edition (JEE)* apresenta facilidades em recursos computacionais e altamente recomendadas para o desenvolvimento de aplicações distribuídas. Apresenta uma vasta *API* especificada pela *Sun*, proporcionando padrões de implementação separados por camadas e cada camada possui seus componentes e serviços específicos e privados por um container. As camadas serão apresentadas nos tópicos seguintes (CAELUM, 2013).

- **CAMADA WEB:** camada implementada por *JSPs* e *Servlets* interagindo com a camada cliente fornecendo serviços de apresentação. O *Servlet* controla as ações do usuário na página de apresentação e a tecnologia *JSP* fornece algumas lógicas de apresentação da página web. Esta camada reside em um *Web container* disponível em servidores como o *TomCat*.
- **CAMADA CLIENTE:** possui três tipos de containers, *Applet container*, *HTML container* e *Application Client*. O acesso a essa camada se dá através de interfaces disponíveis em cada container. O *HTML container* é apresentado através de *Web browser* disponibilizando funcionalidades e recursos para o usuário. A *Application Client* fornece serviços para execução de classes *stand-alone* para interação com o sistema e o *Applet container* fornece serviços para a execução de componentes *applet* tornando funcional para o usuário.
- **CAMADA DE NEGÓCIOS:** nesta camada são tratadas as lógicas de regras de negócio da aplicação, implementando de alocação de recursos, persistência, mapeamento de transações, validação de dados e segurança.
- **CAMADAS EIS:** camada de interação com banco de dados, sistemas de legado e integração com outros sistemas não *JEE*. Sua abreviatura se dá

pelo nome *Enterprise Information System* (Sistemas de Informações Empresariais).

A plataforma é voltada para servidores abstraindo camadas complexas do servidor como implantação, gerenciamento de aplicativos, tornando a plataforma escalável e de alta disponibilidade. Uma das características dessa tecnologia são as anotações para reduzir a quantidade de arquivos *XML*, elevando a produtividade e facilitando o desenvolvimento (CAELUM, 2013), (TEMPLE et al., 2004) e (DEV MEDIA, 2013).

2.3 – JAVA SERVER FACES (JSF)

O *framework Java Server Faces (JSF)* proporciona um desenvolvimento ágil, a arquitetura nos fornece conjuntos de componentes para interface de usuário pré-desenvolvidas e modelos de componentes que permitem a integração de componentes adicionais. A arquitetura fornece também uma gama de tratamento necessário para manipulação de eventos e organização devido à utilização do padrão *Model, View, Controller (MVC)*. O *Framework* é responsável apenas pela apresentação visual, fornecendo toda a lógica da aplicação e de negócios, o mapeamento do objeto relacional, serviços *Web* entre outros serviços não pertencem ao escopo do *JSF*. Nos tópicos seguintes descrevemos alguns pontos chave do *Framework JSF* (LOCKOW; MELO, 2010), (GOMES, 2008) e (HORSTMANN; DAVID, 2005).

- **ARQUITETURA MVC:** conceito de desenvolvimento separado por três camadas: *Model*, classe que trabalha com o armazenamento e busca de dados; *View*, responsável por apresentar os dados resultantes do *Model*. O *Controller* é responsável por responder às ações do usuário, trabalha sobre os dados fornecidos pelo *Model*.
- **CONVERSÃO DE DADOS:** o *JSF* trata todo o conteúdo digitado nos formulários web, pois os dados são do tipo *string*, logo o *JSF* faz a conversão para o tipo necessário.

- VALIDAÇÃO E MANIPULAÇÃO DE ERROS: fácil implementação de validação de campos, gerando restrições de campos como preenchimento obrigatório ou até mesmo o tipo obrigatório de dados a ser inserido.
- INTERNACIONALIZAÇÃO: o *JSF* gerencia questões como a de codificação e seleção de *resource bundles*.
- COMPONENTES CUSTOMIZADOS: fácil acoplamento de novos componentes.
- REDERIZADORES ALTERNATIVOS: o padrão do *JSF* é gerar comandos para página *HTML*, porém pode-se estender o framework para outras linguagens.

O *Java Server Faces (JSF)* é uma especificação para um *framework* de componentes de desenvolvimento *Web Java*, definida através da *Java Community Process (JCP)*, entidade que especifica a evolução da linguagem *Java* de acordo com o mercado. Essa especificação significa que grandes empresas participaram e aprovou sua definição logo a tornando um padrão de mercado.

O *framework JSF* possui implementações para o uso de todos os recursos padrão. Tais implementações são *Sum Mojarra* e *MyFaces* da *Apache*. O fato de ser um padrão de mercado leva as empresas a desenvolverem bibliotecas de componentes adicionais como *ICEFaces*, *RichFaces*, *PrimeFaces* e o *PrimeFaces Mobile* (LOCKOW; MELO, 2010), (GOMES, 2008) e (HORSTMANN; DAVID, 2005).

2.4 – JAVA PERSISTENCE API (JPA)

O *Java Persistence API (JPA)* é uma parte da especificação *Enterprise Java Beans (EJB)* que implementa tecnologias *Object Relational Mapping (ORM)*, responsável pelo mapeamento objeto-relacional entre as classes de entidades do sistema e as tabelas da base de dados, tornando transparente a gravação, leitura, atualização e exclusão dos objetos para o banco de dados relacionais sem a necessidade de códigos *Java Database Connectivity (JDBC)*. Sendo assim, podemos gerenciar operações de manipulação de dados para um *framework* de persistência que

implementa a *API JPA*, *frameworks* como (*Hibernate*, *TopLink* e *EclipseLink*) (CAELUM, 2013), (LOCKOW; MELO, 2010) e (DEVMEDIA, 2013) .

A grande vantagem do *JPA* é:

- PORTABILIDADE DE *FRAMEWORKS* e *SGBD*: pode-se trocar de *framework* de persistência e sistemas gerenciadores de banco de dados (*SGBD*) sem se preocupar com o código da aplicação devido à especificação *JPA*.

A especificação *JPA* tem como principal objetivo simplificar a persistência com o banco de dados relacional (CAELUM, 2013), (LOCKOW; MELO, 2010) e (DEVMEDIA, 2013) .

2.5 – BANCO DE DADOS MySQL

O Sistema gerenciador de Banco de Dados (*SGBD*) *MySQL* é a ferramenta *Open Source* mais usada entre as grandes empresas devido a sua agilidade, confiabilidade e simplicidade. Seu desenvolvimento tem como objetivo suportar grande fluxo de dados de maneira rápida e seguro, suportando ambientes de alta produção e demanda. O sistema cliente/servidor é multitarefa fornecendo recursos para diversos acessos simultâneos.

Sua plataforma fora desenvolvida na linguagem C e C++ disponibilizando características robustas como algumas citadas seguintes:

- *MULT-THREADS*
- PORTABILIDADE
- SEGURANÇA
- ESCALABILIDADE
- CONECTVIDADE

Devido a essas características, é possível afirmar que o *MySQL* é um servidor de banco de dados robusto muito rápido, disponibilizando serviços multitarefa e multiusuários destinados à produção de alto volume de dados. O fato de sua

conectividade, velocidade e segurança ser de alto nível torna o *MySQL* muito recomendado para o uso em aplicações *Web* (MYSQL, 2013).

2.6 – FRAMEWORK PRIMEFACES

O *framework PrimeFaces* é uma biblioteca de componentes leve para desenvolvimento *Web* baseados na API padrão do *JSF*. Por padrão seus componentes foram construídos para trabalhar com *AJAX* sendo composta por cerca de 90 componentes de interface *Web* com código aberto. Esta biblioteca foi uma das primeiras a dar suporte ao *JSF2.0*, sendo sua utilização voltada a *layouts* ricos. O *PrimeFaces* permite a integração de novos componentes ao seu conjunto por meio de uma especificação *JSF*, sendo organizada em três módulos, representados na figura 1: (PRIMEFACES, 2013), (ÇIVICI, 2013) e (SANTOS, 2013).



Figura 3 Módulos PrimeFaces (SANTOS, 2013).

- *Modulo Ui Componentes:* São componentes com funcionalidades encapsuladas de *AJAX*, *JavaScript* e gráficos.
- *Optimos:* modulo oferece soluções para facilitar o desenvolvimento *JSF*, abstraindo a camada de declarações de *XML* e extensões de segurança.
- *Faces Trace:* modulo responsável por melhorar a rastreabilidade de aplicações baseadas em *JSF* além de coletar informações sobre desempenho.

Além do módulo adicional denominado *Touch Faces*, que fornece um conjunto de componentes para desenvolvimento *Web Mobile* (PRIMEFACES, 2013), (ÇIVICI, 2013) e (SANTOS, 2013).

2.6.1 – PrimeFaces Mobile

O *framework PrimeFaces Mobile* possibilita a integração de vários dispositivos em um único sistema através de um Kit de renderização (*Mobile RenderKit*) de interface de usuário para implementação de páginas *JSF* que são otimizadas para dispositivos moveis. Este *framework* tem seu desenvolvimento baseado no *jQuery Mobile* e devido a isso o *PrimeFaces Mobile* suporta uma vasta lista de dispositivos, sua interface se ajusta ao tamanho da tela do dispositivo, podendo ser utilizado em *smartphones*, *tablets* e até mesmo no próprio Notebook. Sendo assim, podemos afirmar que é possível a integração de um mesmo sistema com vários dispositivos, todos acessando o mesmo servidor de aplicação (ÇIVICI, 2013).

3 – ANÁLISE E ESPECIFICAÇÃO DO SISTEMA

Neste capítulo serão apresentadas as ferramentas utilizadas para análise, especificações de requisitos funcionais e a modelagem de diagramas para melhor entendimento do sistema de gestão de ordem de serviços e vistas GOSV.

3.1 – MAPA MENTAL

Para melhor abstração do sistema foi desenvolvido um Mapa Mental ou *Mind Map*, um diagrama responsável por facilitar o entendimento de uma ideia lógica assim como organizar tópicos relacionados a partir de um determinado foco inicial (SOUZA, 2004). O conceito de mapa mental foi desenvolvido por Tony Buzan na década de 70 e seu uso, neste projeto de conclusão de curso, contribui para a definição das funcionalidades do sistema a ser desenvolvido.

A Figura 3 ilustra as atividades do sistema GOSV:

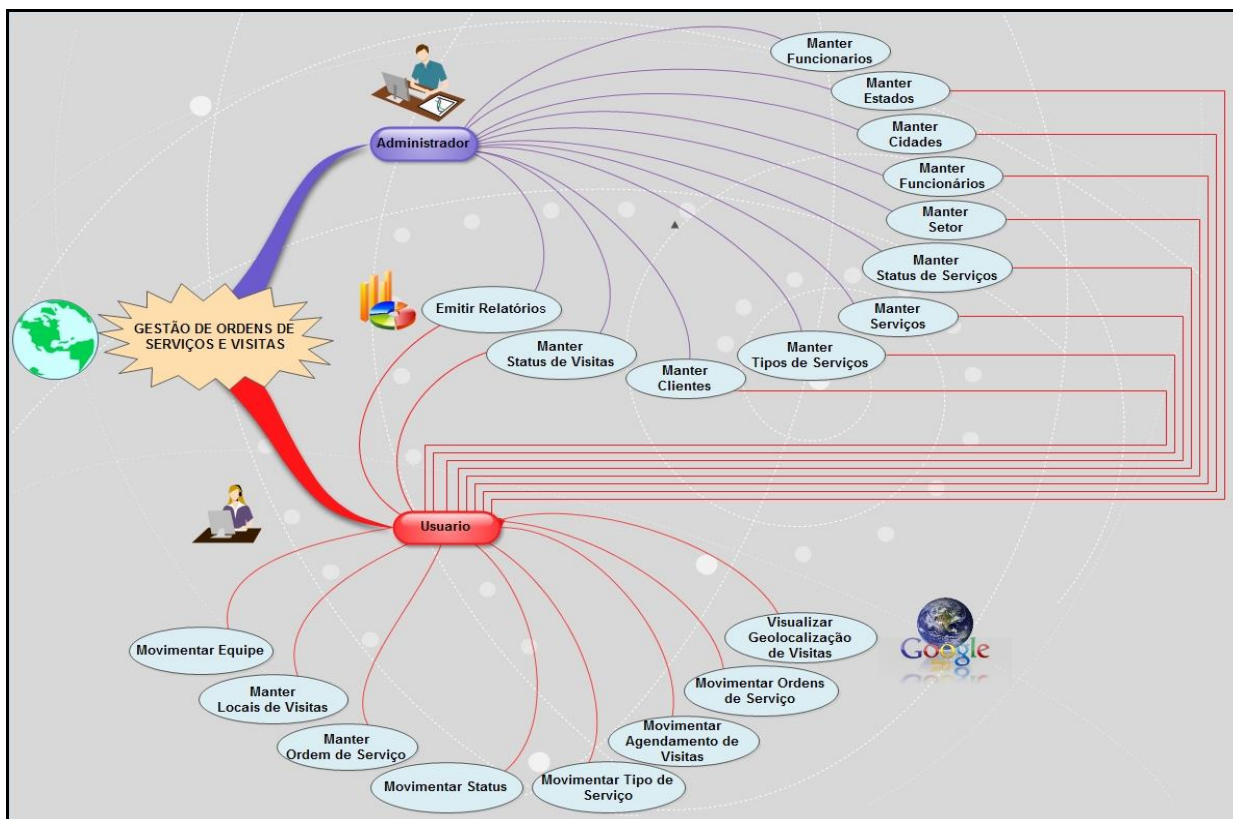


Figura 4 – Mapa Mental – Aplicação GOSV

O mapa mental apresentado fornece uma análise lógica sobre as funcionalidades de seus respectivos atores do sistema GOSV.

3.2 – LISTA DE REQUISITOS

Entre diversas funcionalidades da aplicação GOSV, destacam-se os seguintes requisitos:

- Cadastrar funcionários;
- Cadastrar clientes;
- Cadastrar empresas;
- Cadastrar setores;
- Cadastrar serviços;
- Cadastrar tipos de serviços;
- Cadastrar estados;
- Cadastrar cidades;
- Cadastrar unidades de medidas;
- Cadastrar *status* de visitas;
- Cadastrar *status* de serviços;
- Cadastrar localização de visitas
- Cadastrar localização de serviços
- Movimentar equipes;
- Movimentar ordens de serviço;
- Movimentar agendamento de visitas;
- Movimentar/visualizar geolocalização de visitas;
- Emitir relatório de clientes, funcionários, ordens de serviços, visitas e outros.

3.3 – DIAGRAMA E ESPECIFICAÇÃO DE CASOS DE USO

Para melhor descrever as funcionalidades da aplicação, foram elaborados alguns diagramas como parte da documentação que descreve a interação do usuário com o sistema, dispondo de narrativas para especificação dos diagramas descrevendo em forma textual a interação do usuário e possíveis variantes do sistema. Como elemento primário da UML, utiliza-se o Diagrama de Casos de Uso para especificar o comportamento de um sistema ou de parte de um sistema e é uma descrição de um conjunto de seqüências de ações para completar um determinado processo (BOOCH; JACOBSON; RUMBAUCH, 2005).

A Figura 4 apresenta o Diagrama de Caso de Uso para a entidade “Administrador”:

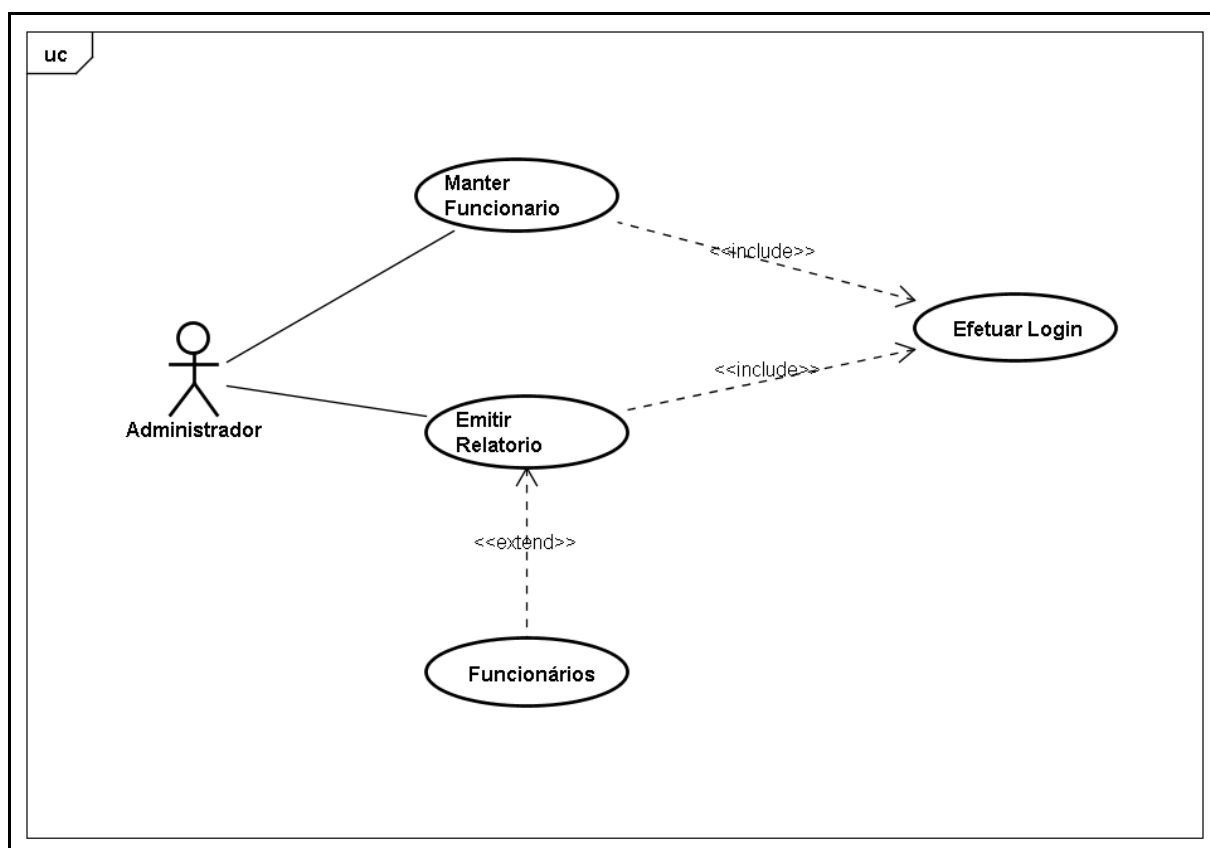


Figura 5 – Diagrama de Caso De Uso – Administrador

A Figura 6 apresenta o Diagrama de Caso de Uso comum para as entidades “Administrador” e “Usuário” com representação “Manter”:

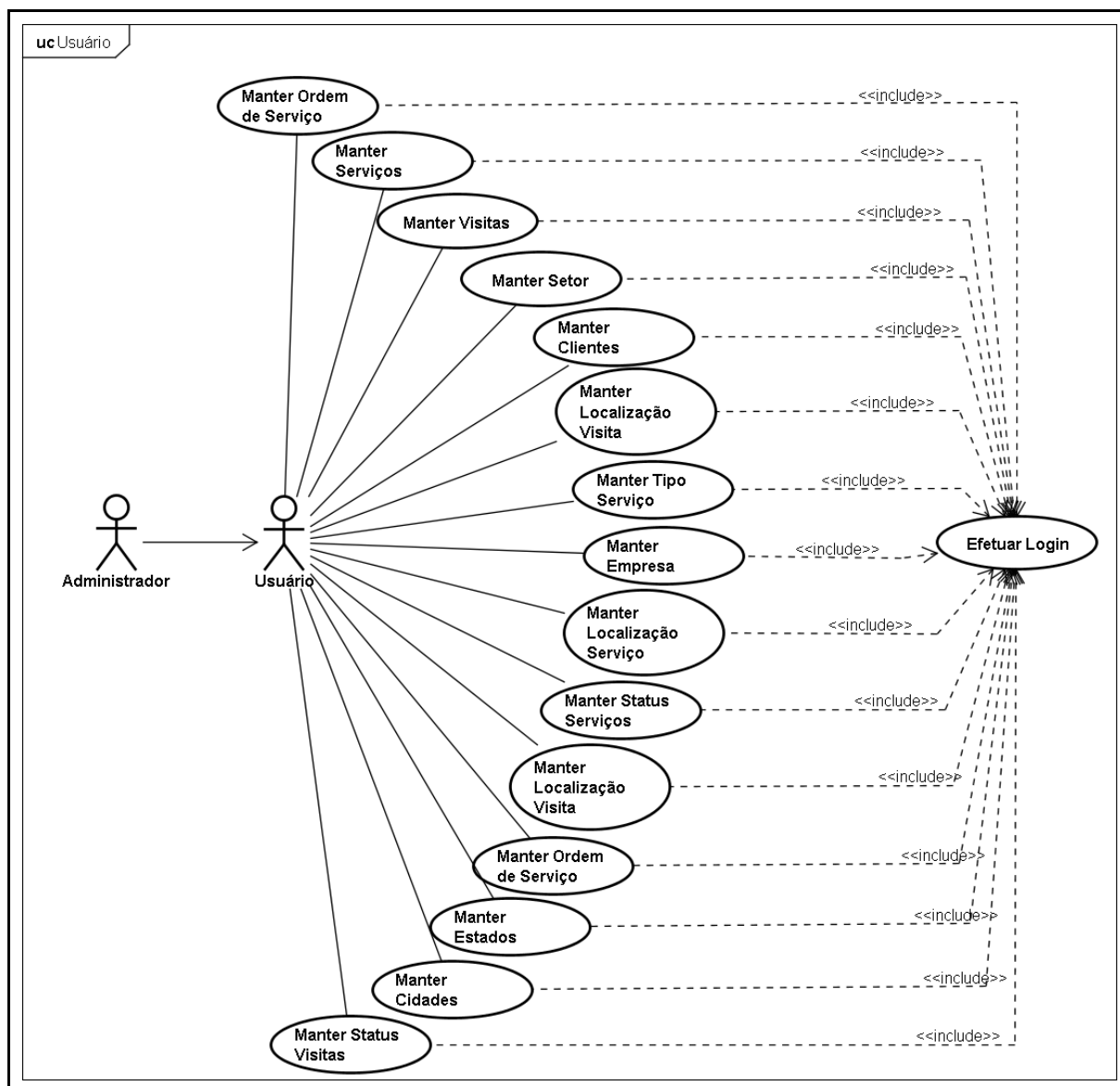


Figura 6 – Diagrama de Caso de Uso comum – Administrador e Usuário

A Figura 7 apresenta o Diagrama de Caso de Uso comum para as entidades “Administrador” e “Usuário” com representação “Movimentar e Emitir”:

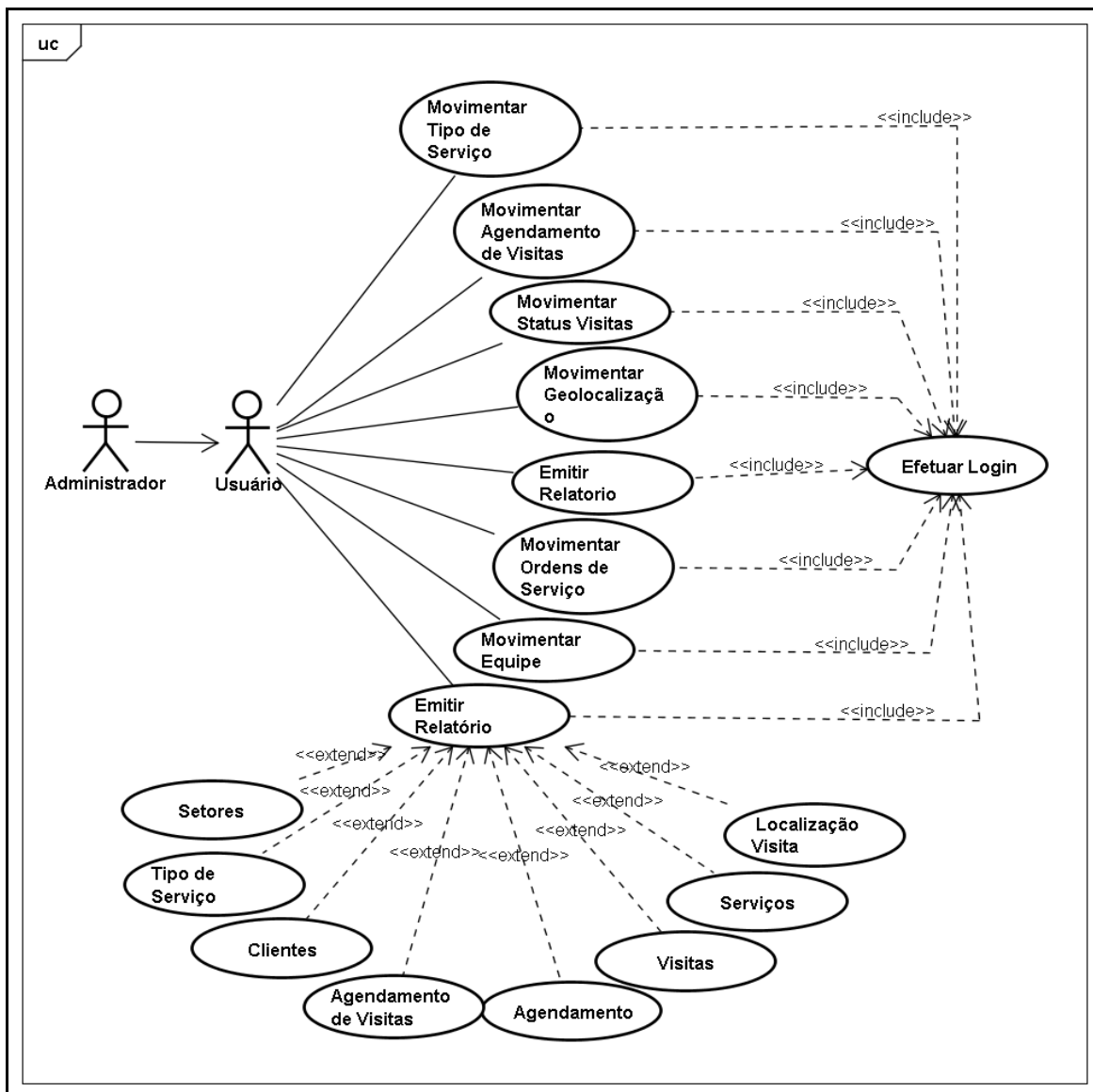


Figura 7 – Diagrama de Caso de Uso comum – Administrador e Usuário

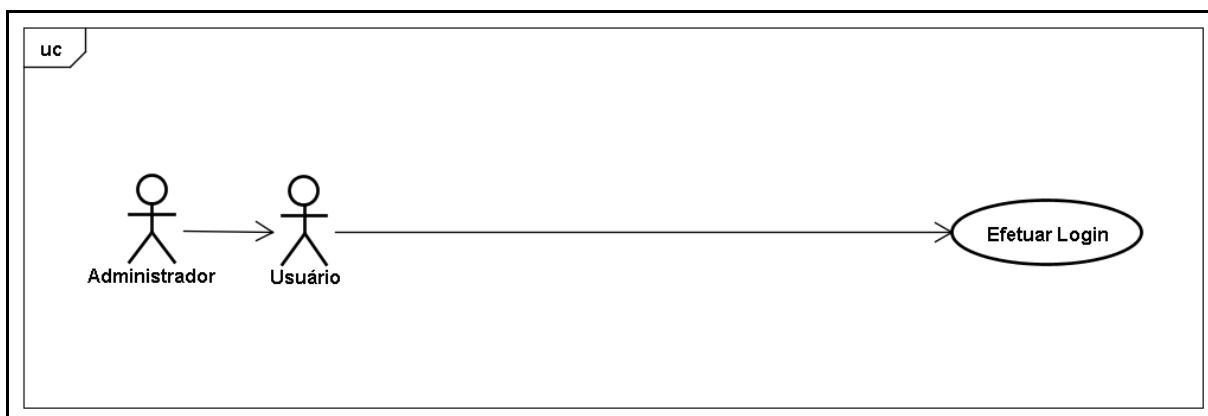


Figura 8 UC1 – Diagrama de Caso de Uso Efetuar Login

Nome do Caso de uso	Efetuar Login
Atores	Administrador e Usuário
Pré-Condições	Não existe
Cenário Principal	<ol style="list-style-type: none"> 1- O sistema solicita os dados para efetuar <i>login</i>. 2- O usuário ou administrador informa os dados. 3- O usuário ou administrador confirma. 4- O sistema recupera os dados informados 5- O sistema valida os dados 6- O usuário se conecta no sistema
Cenário Alternativo	Não existe
Casos de Teste	<ol style="list-style-type: none"> 1- Caso os dados estejam corretos executa a operação 2- Caso os dados não estejam corretos cancela a operação

Tabela 1 – Efetuar Login

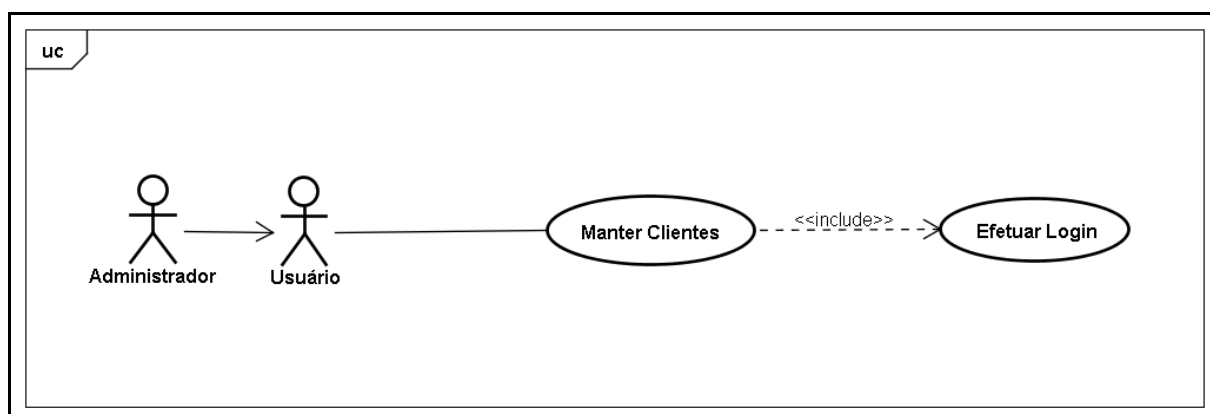


Figura 9 UC2 – Diagrama de Caso de Uso Manter Cliente

Nome do Caso de uso	Manter Cliente
Atores	Administrador e Usuário
Pré-Condições	Efetuar Login
Cenário Principal	<ol style="list-style-type: none"> 1- Usuário ou Administrador Informa os dados do cliente 2- O sistema valida as informações 3- O sistema cadastra o cliente
Cenário Alternativo	Usuário ou administrador pode cancelar a operação durante o processo
Casos de Teste	1- Caso os campos informados estejam corretos finaliza a

	operação 2- Caso os campos não estejam corretos emite mensagem de alerta
--	---

Tabela 2 – Manter Cliente

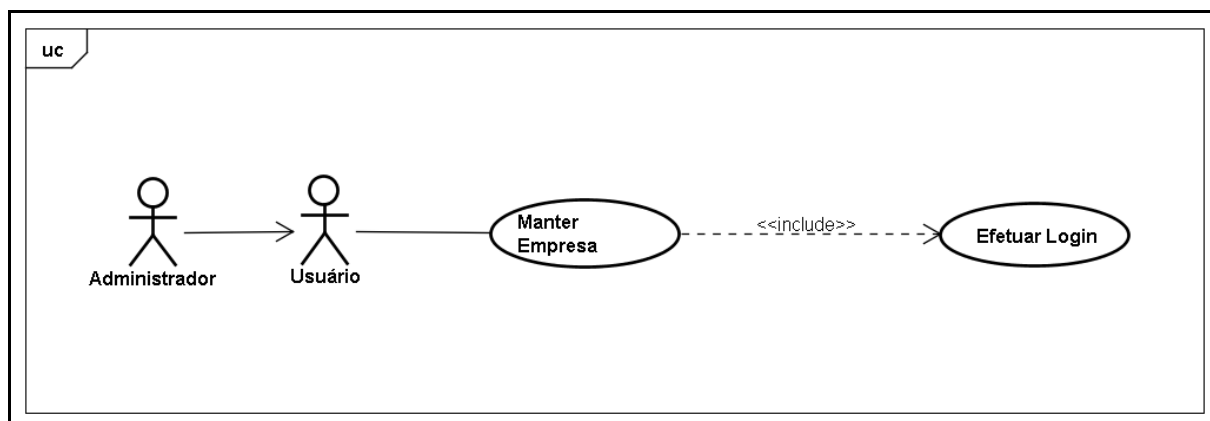


Figura 10 UC3 – Diagrama de Caso de Uso Manter Empresa

Nome do Caso de uso	Manter Empresa
Atores	Administrador e Usuario
Pré-Condições	Efetuar Login
Cenário Principal	1- Usuario ou Administrador Informa os dados da empresa 2- O sistema valida as informações 3- O sistema cadastra a empresa
Cenário Alternativo	Usuário ou administrador pode cancelar a operação durante o processo
Casos de Teste	1- Caso os campos informados estejam corretos finaliza a operação 2- Caso os campos não estejam corretos emite mensagem de alerta

Tabela 3 – Manter Empresa

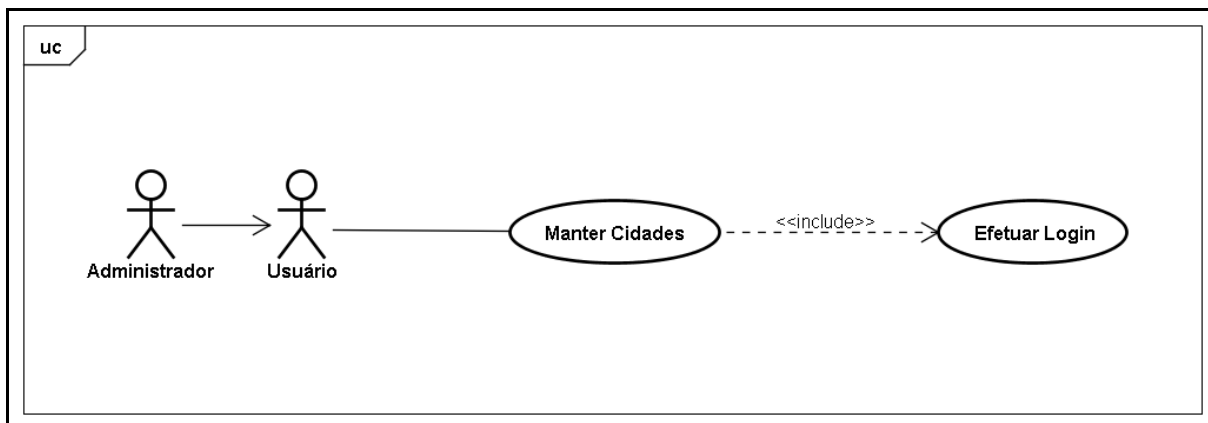


Figura 11 UC4 – Diagrama de Caso de Uso Manter Cidades

Nome do Caso de uso	Manter Cidades
Atores	Administrador e Usuario
Pré-Condições	Efetuar Login
Cenário Principal	<ol style="list-style-type: none"> 1- Usuario ou Administrador Informa os dados da cidade 2- O sistema valida as informações 3- O sistema cadastra a cidade
Cenário Alternativo	Usuário ou administrador pode cancelar a operação durante o processo
Casos de Teste	<ol style="list-style-type: none"> 1- Caso os campos informados estejam corretos finaliza a operação 2- Caso os campos não estejam corretos emite mensagem de alerta

Tabela 4 – Manter Cidades

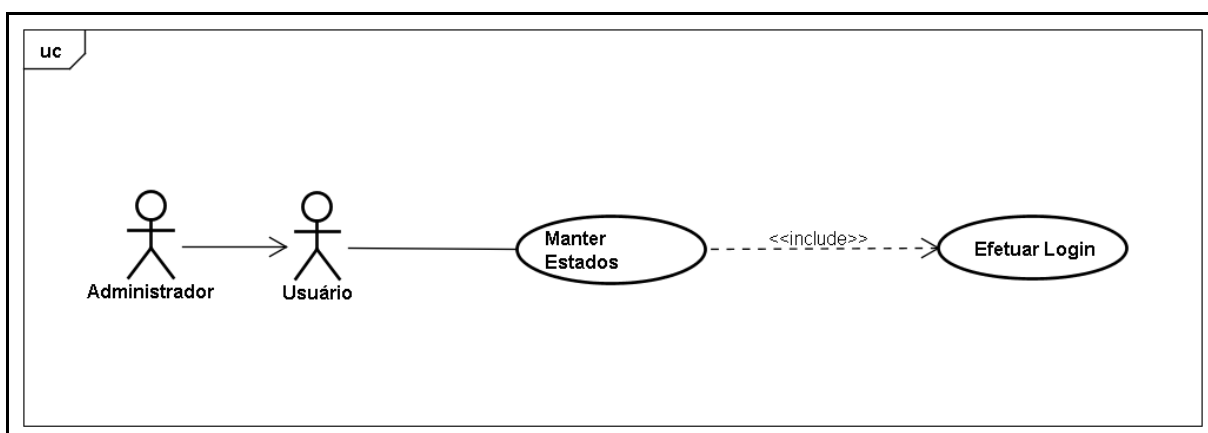


Figura 12 UC5 – Diagrama de Caso de Uso Manter Estados

Nome do Caso de uso	Manter Estados
Atores	Administrador e Usuario
Pré-Condições	Efetuar Login
Cenário Principal	<ol style="list-style-type: none"> 1- Usuario ou Administrador Informa os dados do estado 2- O sistema valida as informações 3- O sistema cadastra o estado
Cenário Alternativo	Usuário ou administrador pode cancelar a operação durante o processo
Casos de Teste	<ol style="list-style-type: none"> 1- Caso os campos informados estejam corretos finaliza a operação 2- Caso os campos não estejam corretos emite mensagem de alerta

Tabela 5 – Manter Estados

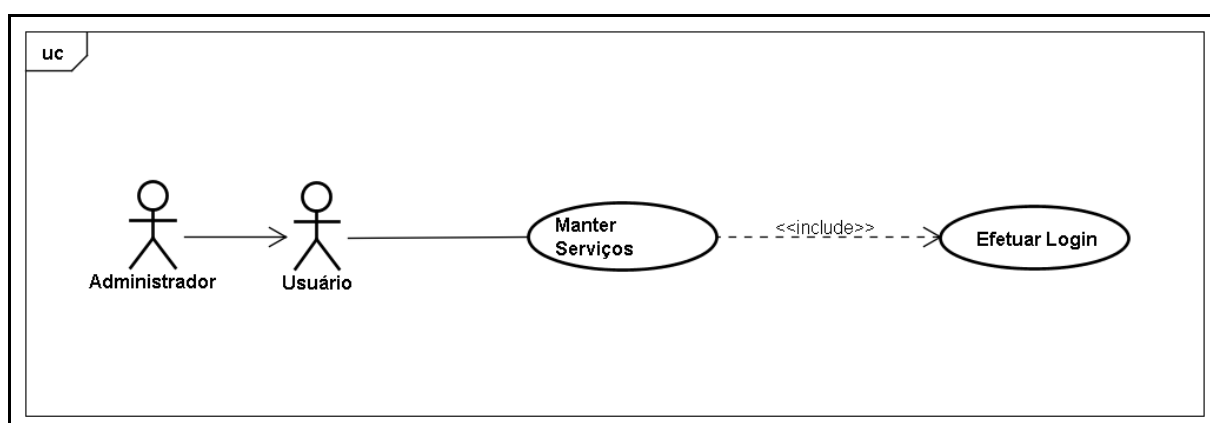


Figura 13 UC6 – Diagrama de Caso de Uso Manter Serviços

Nome do Caso de uso	Manter Serviços
Atores	Administrador e Usuario
Pré-Condições	Efetuar Login
Cenário Principal	<ol style="list-style-type: none"> 1- Usuario ou Administrador Informa os dados da Serviço 2- Usuario ou Administrador verifica e informa o Funcionario disponivel 3- O sistema valida as informações 4- O sistema cadastra o serviço
Cenário Alternativo	Usuário ou administrador pode cancelar a operação durante o processo Caso o funcionario não estiver disponivel deve-se efetuar o agendamento

Casos de Teste	<ol style="list-style-type: none"> 1- Caso os campos informados estejam corretos finaliza a operação 2- Caso os campos não estejam corretos emite mensagem de alerta
-----------------------	--

Tabela 6 – Manter Serviços

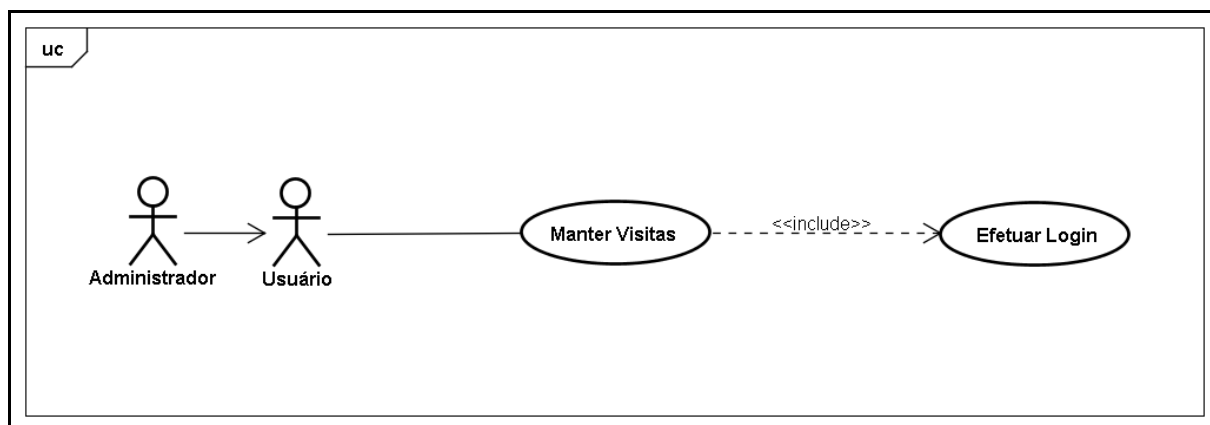


Figura 14 UC7 – Diagrama de Caso de Uso Manter Visitas

Nome do Caso de uso	Manter Visitas
Atores	Administrador e Usuario
Pré-Condições	Efetuar Login
Cenário Principal	<ol style="list-style-type: none"> 1- Usuario ou Administrador Informa os dados da Visita 2- Usuario ou Administrador verifica e informa o Funcionario disponivel 3- O sistema valida as informações 4- O sistema cadastra a Visita
Cenário Alternativo	<p>Usuário ou administrador pode cancelar a operação durante o processo</p> <p>Caso o funcionario não estiver disponivel deve-se efetuar o agendamento</p>
Casos de Teste	<ol style="list-style-type: none"> 1- Caso os campos informados estejam corretos finaliza a operação 2- Caso os campos não estejam corretos emite mensagem de alerta

Tabela 7 – Manter Visitas

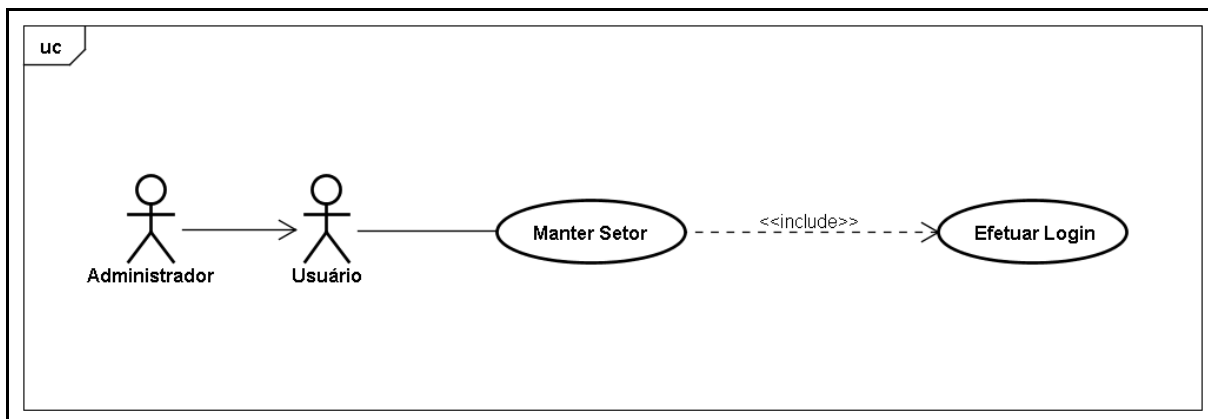


Figura 15 UC8 – Diagrama de Caso de Uso Manter Setor

Nome do Caso de uso	Manter Setor
Atores	Administrador e Usuario
Pré-Condições	Efetuar Login
Cenário Principal	<ol style="list-style-type: none"> 1- Usuario ou Administrador Informa os dados do setor 2- O sistema valida as informações 3- O sistema cadastra o setor
Cenário Alternativo	Usuário ou administrador pode cancelar a operação durante o processo
Casos de Teste	<ol style="list-style-type: none"> 1- Caso os campos informados estejam corretos finaliza a operação 2- Caso os campos não estejam corretos emite mensagem de alerta

Tabela 8 – Manter Setor

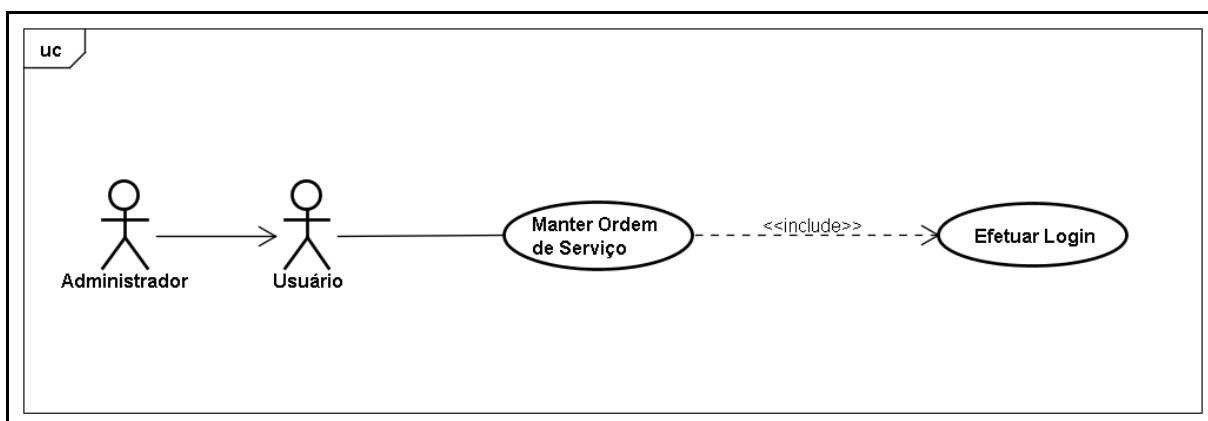


Figura 16 UC9 – Diagrama de Caso de Uso Manter Ordem de Serviço

Nome do Caso de uso	Manter Ordem de Serviço
Atores	Administrador e Usuario
Pré-Condições	Efetuar Login
Cenário Principal	<ol style="list-style-type: none"> 1- Usuario ou Administrador Informa os dados do Serviço 2- Usuario ou Administrador verifica e informa o Funcionario disponivel 3- Usuario ou Administrador informa o tipo de serviço 4- Usuario ou Administrador informa o Cliente 5- O sistema valida as informações 6- O sistema cadastra a ordem de serviço
Cenário Alternativo	<p>Usuário ou administrador pode cancelar a operação durante o processo</p> <p>Caso o funcionario não estiver disponivel deve-se efetuar o agendamento</p> <p>Caso o cliente não esteja cadastrado deve-se efetuar o cadastro</p> <p>Caso o tipo de serviço não esteja cadastrado deve-se efetuar o cadastro</p>
Casos de Teste	<ol style="list-style-type: none"> 1- Caso os campos informados estejam corretos finaliza a operação 2- Caso os campos não estejam corretos emite mensagem de alerta

Tabela 9 – Manter Ordem de Serviço

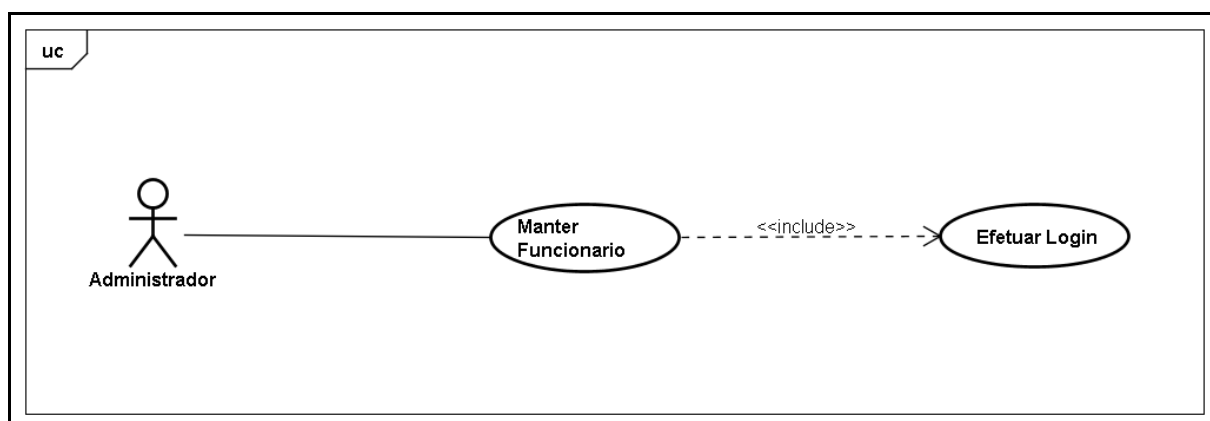


Figura 17 UC10 – Diagrama de Caso de Uso Manter Funcionário

Nome do Caso de uso	Manter Funcionário
Atores	Administrador
Pré-Condições	Efetuar Login
Cenário Principal	<ol style="list-style-type: none"> 1- Administrador Informa os dados do funcionario 2- O sistema valida as informações 3- O sistema cadastra o funcionario
Cenário Alternativo	administrador pode cancelar a operação durante o processo
Casos de Teste	<ol style="list-style-type: none"> 1- Caso os campos informados estejam corretos finaliza a operação 2- Caso os campos não estejam corretos emite mensagem de alerta

Tabela 10 – Manter Funcionário

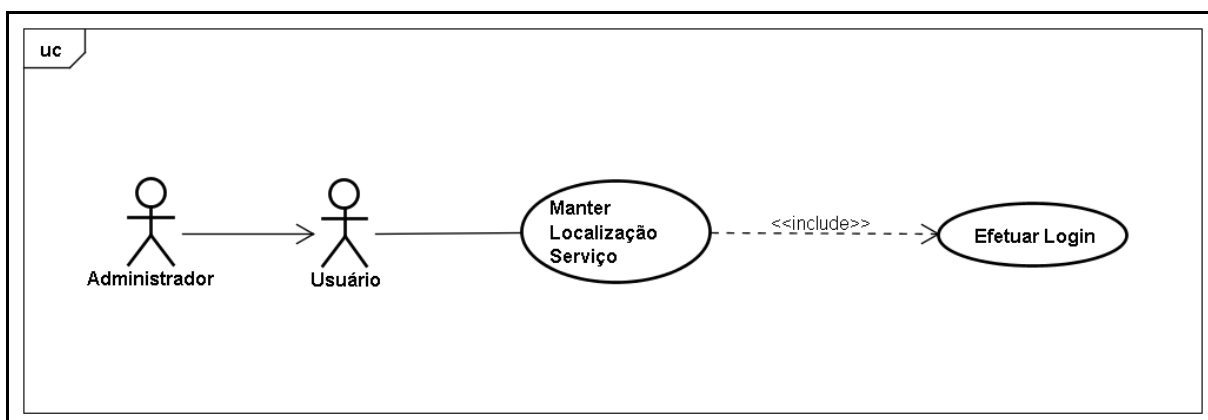


Figura 18 UC11 – Diagrama de Caso de Uso Manter Localização Serviço

Nome do Caso de uso	Manter Localização do Serviço
Atores	Administrador e Usuario
Pré-Condições	Efetuar Login
Cenário Principal	<ol style="list-style-type: none"> 1- Usuario ou Administrador Informa os dados da localização do serviço 2- O sistema verifica os dados no googlemaps 3- O sistema valida as informações 4- O sistema cadastra o localização
Cenário Alternativo	Usuário ou administrador pode cancelar a operação durante o processo

Casos de Teste	<ol style="list-style-type: none"> 1- Caso os campos informados estejam corretos finaliza a operação 2- Caso os campos não estejam corretos emite mensagem de alerta
-----------------------	--

Tabela 11 – Manter Localização do Serviço

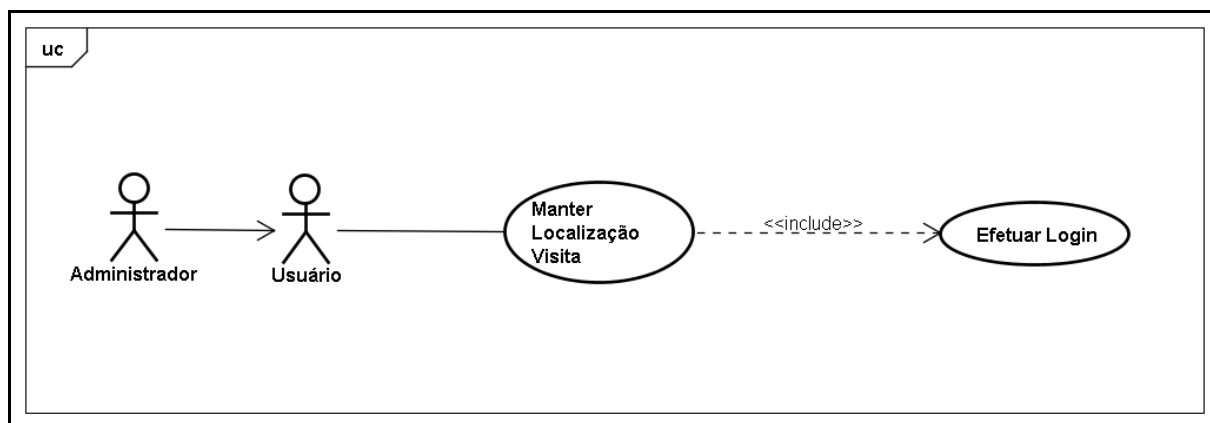


Figura 19 UC12 – Diagrama de Caso de Uso Manter Localização Visita

Nome do Caso de uso	Manter Localização de Visita
Atores	Administrador e Usuario
Pré-Condições	Efetuar Login
Cenário Principal	<ol style="list-style-type: none"> 1- Usuario ou Administrador Informa os dados da localização da visita 2- O sistema verifica os dados no googlemaps 3- O sistema valida as informações 4- O sistema cadastra o localização da visita
Cenário Alternativo	Usuário ou administrador pode cancelar a operação durante o processo
Casos de Teste	<ol style="list-style-type: none"> 1- Caso os campos informados estejam corretos finaliza a operação 2- Caso os campos não estejam corretos emite mensagem de alerta

Tabela 12 – Manter Localização de Visita

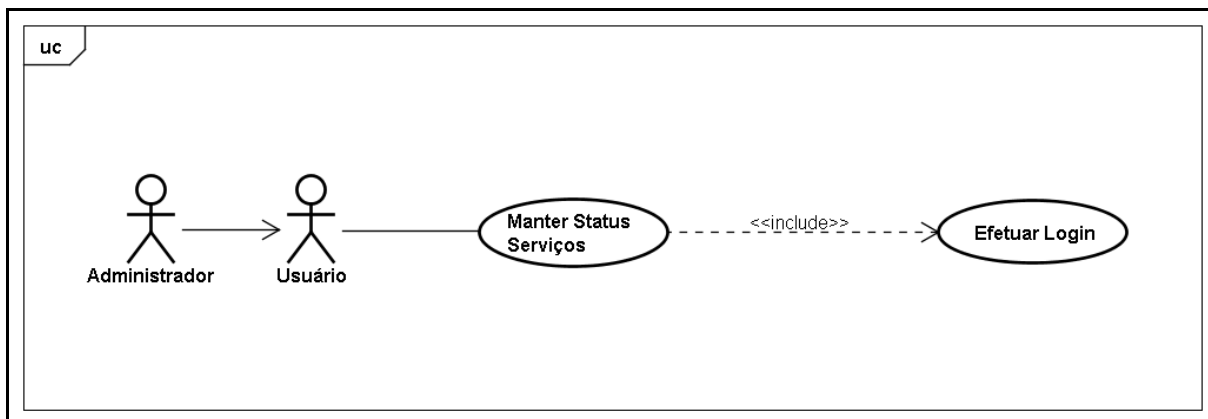


Figura 20 UC13 – Diagrama de Caso de Uso Manter Status Serviço

Nome do Caso de uso	Manter Status Serviço
Atores	Administrador e Usuario
Pré-Condições	Efetuar Login
Cenário Principal	<ol style="list-style-type: none"> 1- Usuario ou Administrador Informa o status do serviço 2- O sistema valida as informações 3- O sistema cadastra o status da visita
Cenário Alternativo	Usuário ou administrador pode cancelar a operação durante o processo
Casos de Teste	<ol style="list-style-type: none"> 1- Caso os campos informados estejam corretos finaliza a operação 2- Caso os campos não estejam corretos emite mensagem de alerta

Tabela 13 – Manter Status Serviço

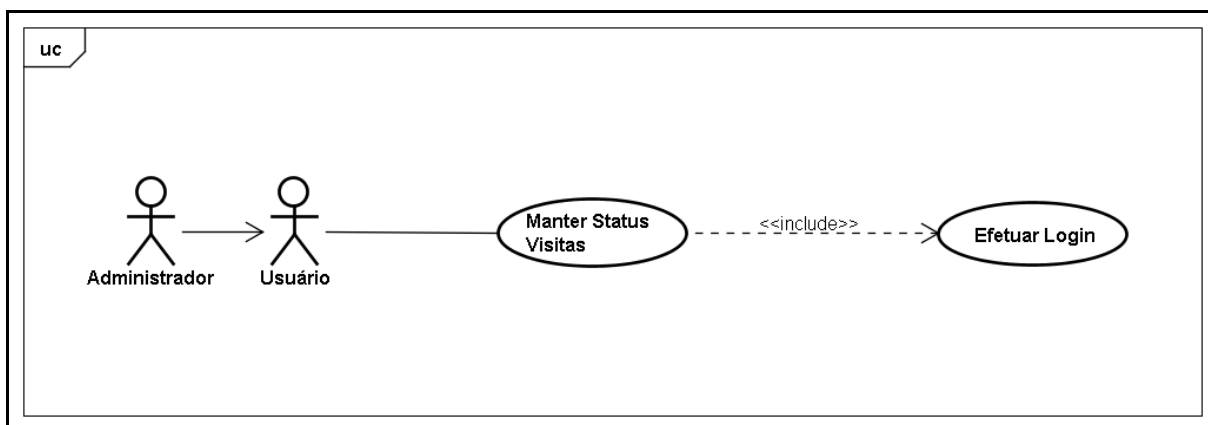


Figura 21 UC14 – Diagrama de Caso de Uso Manter Status Visitas

Nome do Caso de uso	Manter Status Visita
Atores	Administrador e Usuario
Pré-Condições	Efetuar Login
Cenário Principal	<ol style="list-style-type: none"> 1- Usuario ou Administrador Informa o status da visita 2- O sistema valida as informações 3- O sistema cadastra o status da visita
Cenário Alternativo	Usuário ou administrador pode cancelar a operação durante o processo
Casos de Teste	<ol style="list-style-type: none"> 1- Caso os campos informados estejam corretos finaliza a operação 2- Caso os campos não estejam corretos emite mensagem de alerta

Tabela 14 – Manter Status Visita

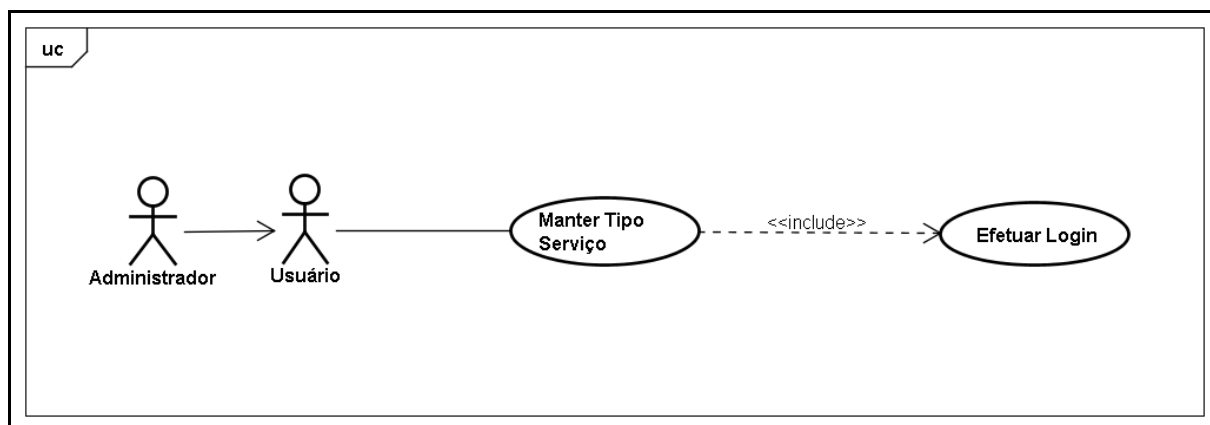


Figura 22 UC15 – Diagrama de Caso de Uso Manter Tipo Serviço

Nome do Caso de uso	Manter Tipo de Serviço
Atores	Administrador e Usuario
Pré-Condições	Efetuar Login
Cenário Principal	<ol style="list-style-type: none"> 1- Usuario ou Administrador Informa o o tipo de serviço 2- O sistema valida as informações 3- O sistema cadastra o tipo de serviço
Cenário Alternativo	Usuário ou administrador pode cancelar a operação durante o processo
Casos de Teste	1- Caso os campos informados

	<p>estajam corretos finaliza a operação</p> <p>2- Caso os campos não estejam corretos emite mensagem de alerta</p>
--	--

Tabela 15 – Manter Tipo Serviço

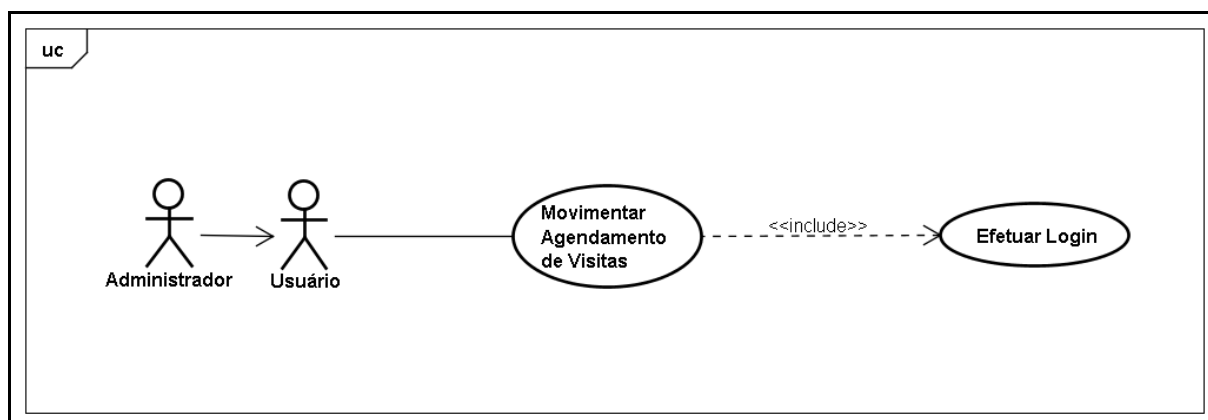


Figura 23 UC16 – Diagrama de Caso de Uso Movimentar Agendamento de Visitas

Nome do Caso de uso	Movimentar Agendamento de Visitas
Atores	Usuário e Administrador
Pré-Condições	Efetuar Login
Cenário Principal	<ol style="list-style-type: none"> 1- Usuário ou Administrador Informa os dados do Agendamento 2- Usuário ou Administrador verifica a disponibilidade 3- O sistema valida os dados informados 4- O usuário seleciona a opção "gravar" 5- O sistema emite uma mensagem de sucesso 6- O sistema cadastra o agendamento
Cenário Alternativo	Usuário ou administrador pode cancelar a operação durante o processo
Casos de Teste	<ol style="list-style-type: none"> 1- Caso os campos informados estajam corretos finaliza a operação 2- Caso os campos não estejam corretos emite mensagem de

	alerta
--	--------

Tabela 16 – Movimentar Agendamento de Visitas

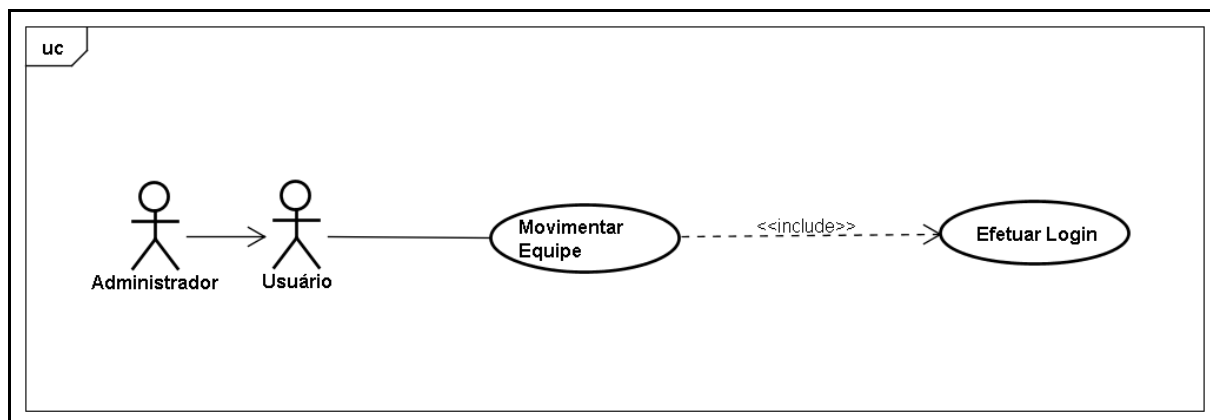


Figura 24 UC17 – Diagrama de Caso de Uso Movimentar Equipe

Nome do Caso de uso	Movimentar Equipe
Atores	Usuário e Administrador
Pré-Condições	Efetuar Login
Cenário Principal	<ol style="list-style-type: none"> 1- Usuário ou Administrador Informa os dados da equipe 2- Usuário ou Administrador verifica a disponibilidade da equipe 3- Usuário ou Administrador Informa os dados do serviço ou visita a ser executado 4- O sistema valida os dados informados 5- O usuário seleciona a opção "gravar" 6- O sistema emite uma mensagem de sucesso 7- O sistema cadastra o serviço ou visita
Cenário Alternativo	Usuário ou administrador pode cancelar a operação durante o processo
Casos de Teste	<ol style="list-style-type: none"> 1- Caso os campos informados estejam corretos finaliza a operação 2- Caso os campos não estejam corretos emite mensagem de alerta

Tabela 17 – Movimentar Equipe

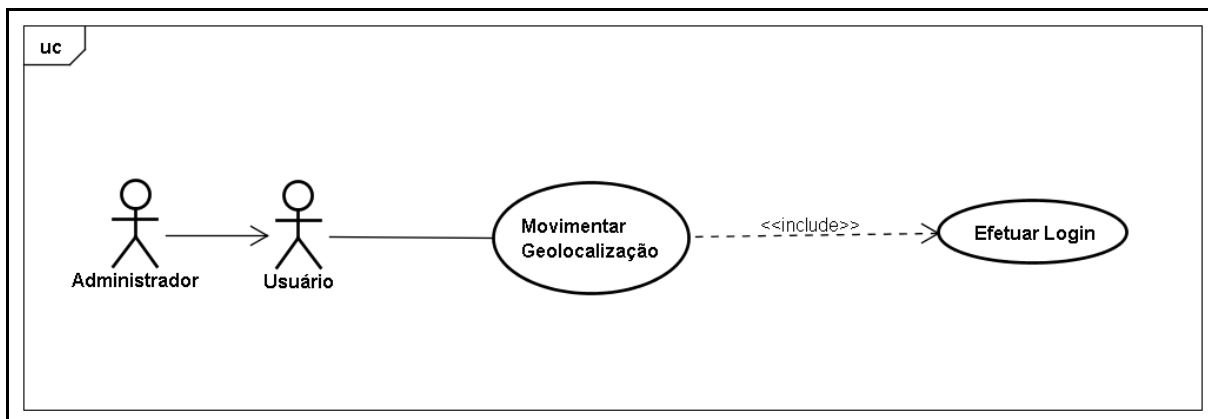


Figura 25 UC18 – Diagrama de Caso de Uso Movimentar Geolocalização

Nome do Caso de uso	Movimentar Geolocalização
Atores	Usuário e Administrador
Pré-Condições	Efetuar Login
Cenário Principal	<ol style="list-style-type: none"> 1- Usuário ou Administrador Informa os dados da geolocalização 2- O sistema valida os dados informados 3- O usuário seleciona a opção “gravar” 4- O sistema emite uma mensagem de sucesso 5- O sistema cadastra a geolocalização
Cenário Alternativo	Usuário ou administrador pode cancelar a operação durante o processo
Casos de Teste	<ol style="list-style-type: none"> 1- Caso os campos informados estejam corretos finaliza a operação 2- Caso os campos não estejam corretos emite mensagem de alerta

Tabela 18 – Movimentar Geolocalização

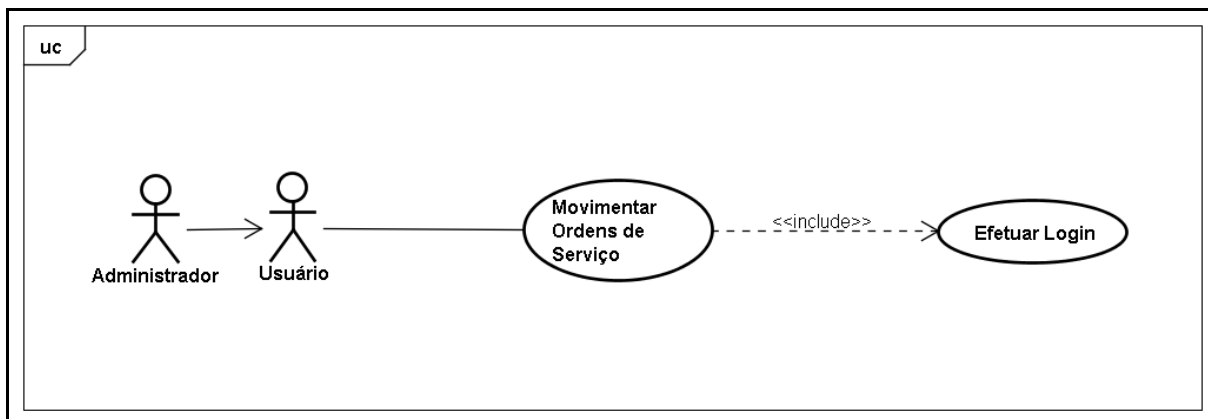


Figura 26 UC19 – Diagrama de Caso de Uso Movimentar Ordem de Serviço

Nome do Caso de uso	Movimentar Ordem de Serviço
Atores	Usuário e Administrador
Pré-Condições	Efetuar Login
Cenário Principal	<ol style="list-style-type: none"> 1- Usuário ou Administrador Informa os dados da ordem de serviço 2- O sistema valida os dados informados 3- O usuário seleciona a opção “gravar” 4- O sistema emite uma mensagem de sucesso 5- O sistema cadastra a ordem de serviço
Cenário Alternativo	Usuário ou administrador pode cancelar a operação durante o processo
Casos de Teste	<ol style="list-style-type: none"> 1- Caso os campos informados estejam corretos finaliza a operação 2- Caso os campos não estejam corretos emite mensagem de alerta

Tabela 19 – Movimentar Ordem de Serviço

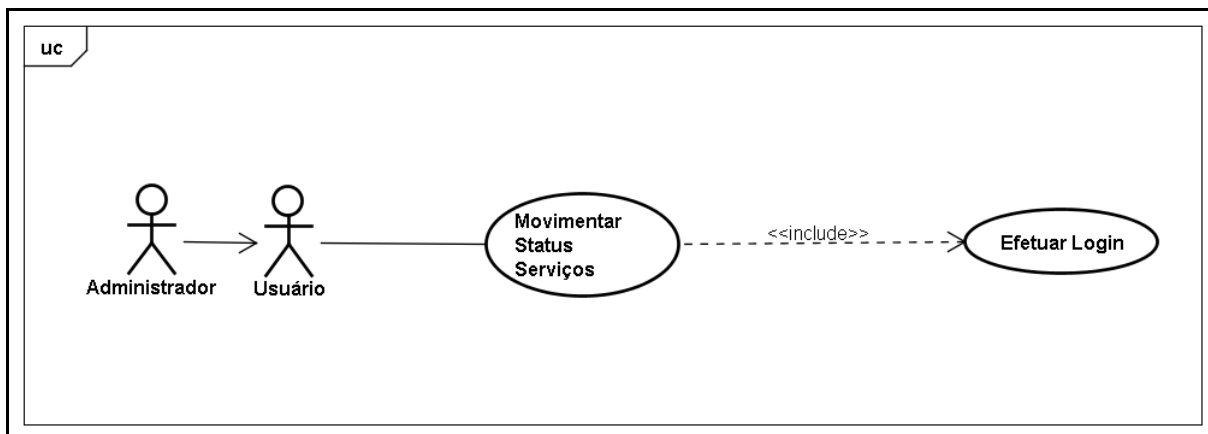


Figura 27 UC20 – Diagrama de Caso de Uso Movimentar Status Serviço

Nome do Caso de uso	Movimentar Status Serviço
Atores	Usuário e Administrador
Pré-Condições	Efetuar Login
Cenário Principal	<ol style="list-style-type: none"> 1- Usuário ou Administrador Informa os dados do status 2- Usuário ou Administrador verifica a situação do serviço 3- O sistema valida os dados informados 4- O usuário seleciona a opção “gravar” 5- O sistema emite uma mensagem de sucesso 6- O sistema cadastra o status do serviço
Cenário Alternativo	Usuário ou administrador pode cancelar a operação durante o processo
Casos de Teste	<ol style="list-style-type: none"> 1- Caso os campos informados estejam corretos finaliza a operação 2- Caso os campos não estejam corretos emite mensagem de alerta

Tabela 20 – Movimentar Status Serviço

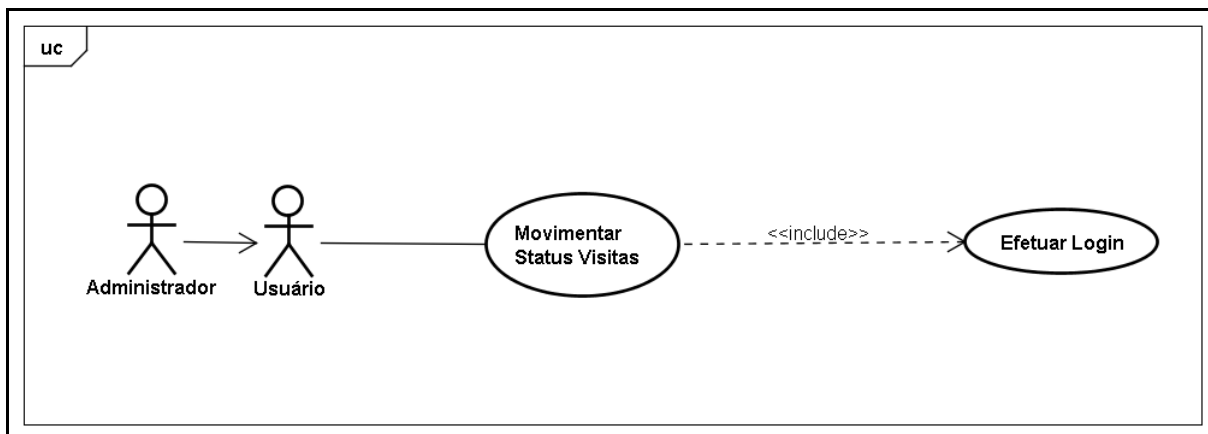


Figura 28 UC21 – Diagrama de Caso de Uso Movimentar Status Visitas

Nome do Caso de uso	Movimentar Status Visitas
Atores	Usuário e Administrador
Pré-Condições	Efetuar Login
Cenário Principal	<ol style="list-style-type: none"> 1- Usuário ou Administrador Informa os dados do status 2- Usuário ou Administrador verifica a situação da visita 3- O sistema valida os dados informados 4- O usuário seleciona a opção "gravar" 5- O sistema emite uma mensagem de sucesso 6- O sistema cadastra o status da visita
Cenário Alternativo	Usuário ou administrador pode cancelar a operação durante o processo
Casos de Teste	<ol style="list-style-type: none"> 1- Caso os campos informados estejam corretos finaliza a operação 2- Caso os campos não estejam corretos emite mensagem de alerta

Tabela 21 – Movimentar Status Visitas

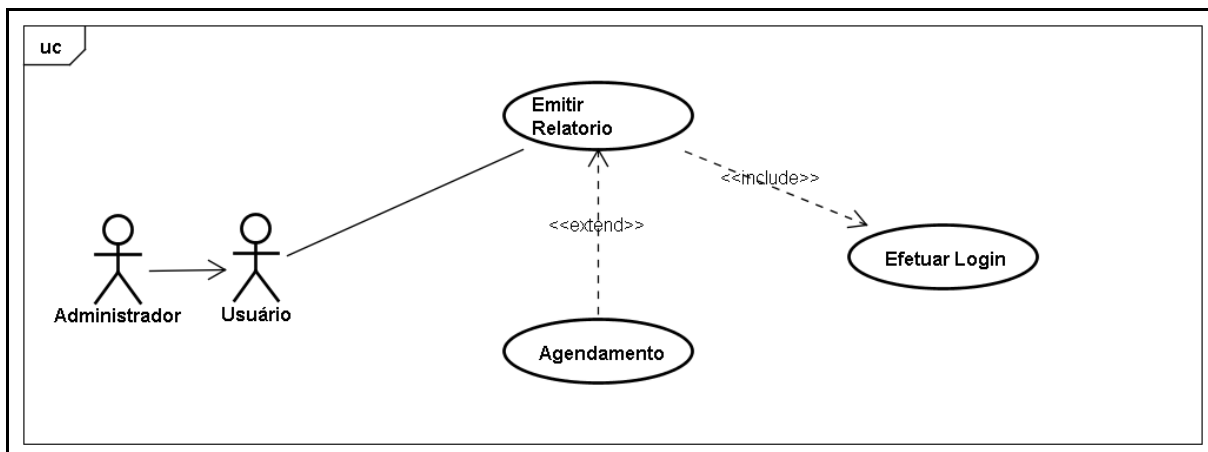


Figura 29 UC22 – Diagrama de Caso de Uso Emitir Relatório de Agendamentos

Nome do Caso de uso	Emitir relatório de agendamentos
Atores	Usuário e Administrador
Pré-Condições	Efetuar Login
Cenário Principal	<ol style="list-style-type: none"> 1- O sistema informa os dados necessarios para relatorio 2- O usuário ou Administrador seleciona a opção “visualizar” 3- O usuário ou Administrador seleciona a opção “imprimir” 4- O sistema imprimi o relatório com sucesso
Cenário Alternativo	Usuário ou administrador pode visualizar o relatorio e não imprimir
Casos de Teste	1- Usuário ou administrador cancela a operação

Tabela 22 – Emitir Relatório de Agendamentos

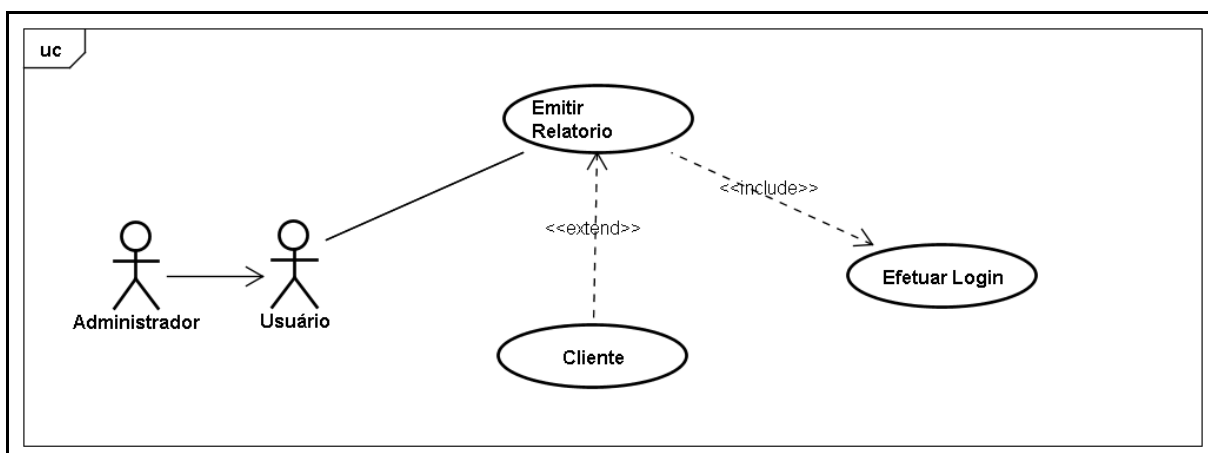


Figura 30 UC23 – Diagrama de Caso de Uso Emitir Relatório de Clientes

Nome do Caso de uso	Emitir relatório de Clientes
Atores	Usuário e Administrador
Pré-Condições	Efetuar Login
Cenário Principal	<p>2- O sistema informa os dados necessarios para relatorio</p> <p>3- O usuário ou Administrador seleciona a opção “visualizar”</p> <p>4- O usuário ou Administrador seleciona a opção “imprimir”</p> <p>5- O sistema imprimi o relatório com sucesso</p>
Cenário Alternativo	Usuário ou administrador pode visualizar o relatorio e não imprimir
Casos de Teste	1- Usuário ou administrador cancela a operação

Tabela 23 – Emitir Relatório de Clientes

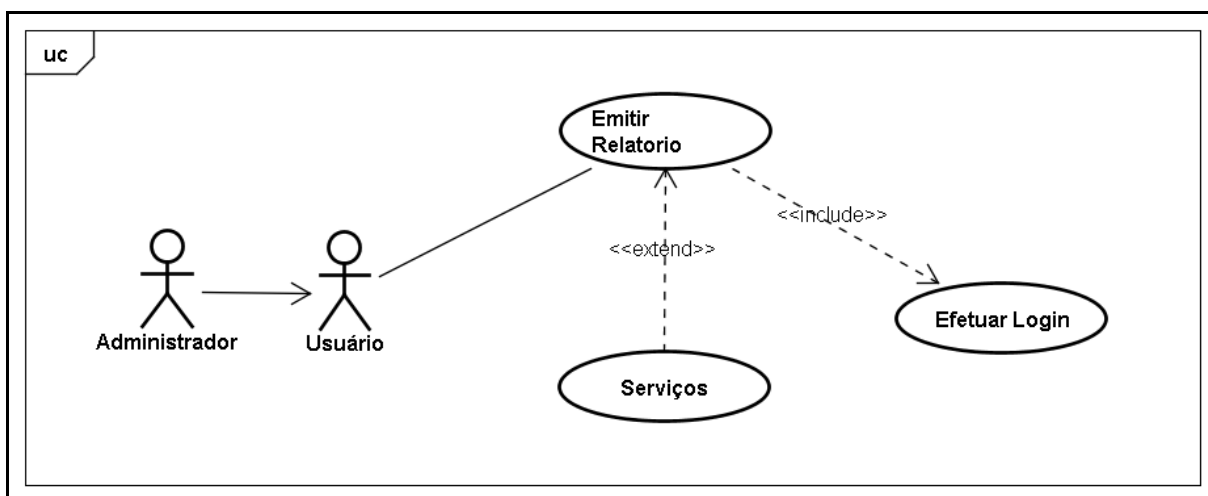


Figura 31 UC24 – Diagrama de Caso de Uso Emitir Relatório de Serviços

Nome do Caso de uso	Emitir relatório de serviços
Atores	Usuário e Administrador
Pré-Condições	Efetuar Login
Cenário Principal	<p>1- O sistema informa os dados necessarios para relatorio</p> <p>2- O usuário ou Administrador seleciona a opção “visualizar”</p> <p>3- O usuário ou Administrador</p>

	seleciona a opção “imprimir” 4- O sistema imprime o relatório com sucesso
Cenário Alternativo	Usuário ou administrador pode visualizar o relatório e não imprimir
Casos de Teste	1- Usuário ou administrador cancela a operação

Tabela 24 – Emitir Relatório de Serviços

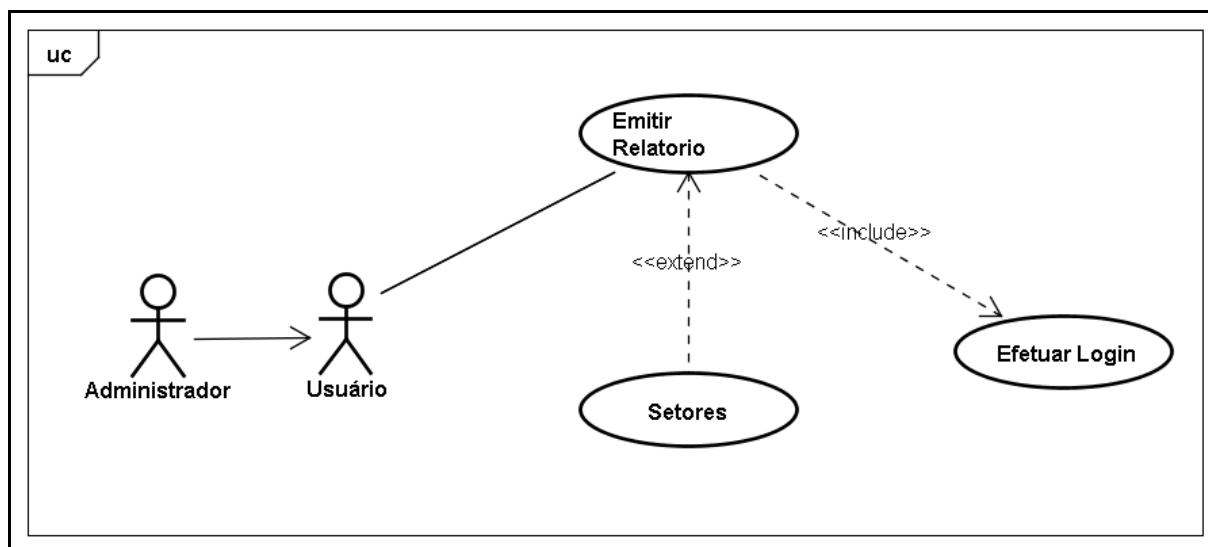


Figura 32 UC25 – Diagrama de Caso de Uso Emitir Relatório de Setores

Nome do Caso de uso	Emitir relatório de setores
Atores	Usuário e Administrador
Pré-Condições	Efetuar Login
Cenário Principal	<ol style="list-style-type: none"> 1- O sistema informa os dados necessários para relatório 2- O usuário ou Administrador seleciona a opção “visualizar” 3- O usuário ou Administrador seleciona a opção “imprimir” 4- O sistema imprime o relatório com sucesso
Cenário Alternativo	Usuário ou administrador pode visualizar o relatório e não imprimir
Casos de Teste	1- Usuário ou administrador cancela a operação

Tabela 25 – Emitir Relatório de Setores

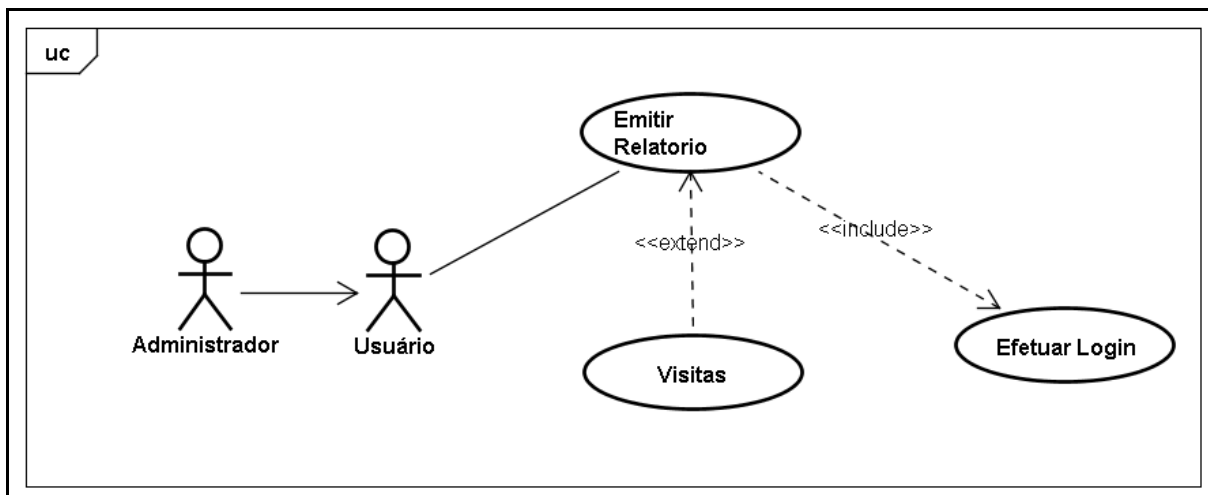


Figura 33 UC26 – Diagrama de Caso de Uso Emitir Relatório de Visitas

Nome do Caso de uso	Emitir relatório de visitas
Atores	Usuário e Administrador
Pré-Condições	Efetuar Login
Cenário Principal	<ol style="list-style-type: none"> 1- O sistema informa os dados necessários para relatório 2- O usuário ou Administrador seleciona a opção “visualizar” 3- O usuário ou Administrador seleciona a opção “imprimir” 4- O sistema imprime o relatório com sucesso
Cenário Alternativo	Usuário ou administrador pode visualizar o relatório e não imprimir
Casos de Teste	<ol style="list-style-type: none"> 1- Usuário ou administrador cancela a operação

Tabela 26 – Emitir Relatório de Visitas

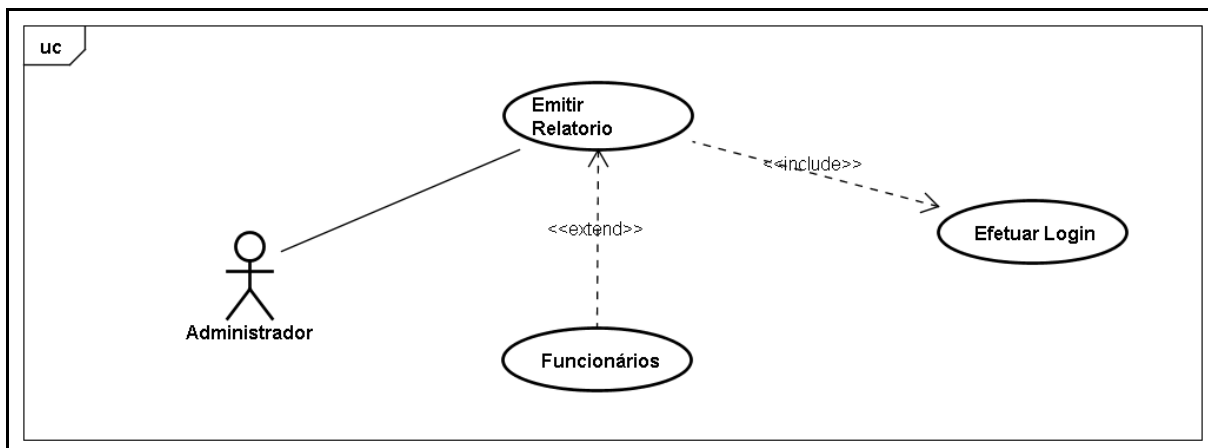


Figura 34 UC27 – Diagrama de Caso de Uso Emitir Relatório de Funcionários

Nome do Caso de uso	Emitir relatório de funcionários
Atores	Administrador
Pré-Condições	Efetuar Login
Cenário Principal	<ul style="list-style-type: none"> 2- O sistema informa os dados necessarios para relatorio 3- O Administrador seleciona a opção “visualizar” 4- O Administrador seleciona a opção “imprimir” 5- O sistema imprimi o relatório com sucesso
Cenário Alternativo	Administrador pode visualizar o relatorio e não imprimir
Casos de Teste	1- Administrador cancela a operação

Tabela 27 – Emitir Relatório de Funcionários

Os diagramas de caso de uso apresentados representam as funcionalidades e suas respectivas narrativas do sistema GOSV para as entidades “Administrador e Usuário”.

3.4 – DIAGRAMA DE ATIVIDADES

O diagrama de atividade é definido pela *UML* para representação do fluxo de controle de uma determinada atividade para a outra de forma que possa ser ilustrada uma modelagem dos aspectos dinâmicos do sistema, logo fornecendo um

entendimento transparente para uma determinação (BOOCH; JACOBSON; RUMBAUCH, 2005).

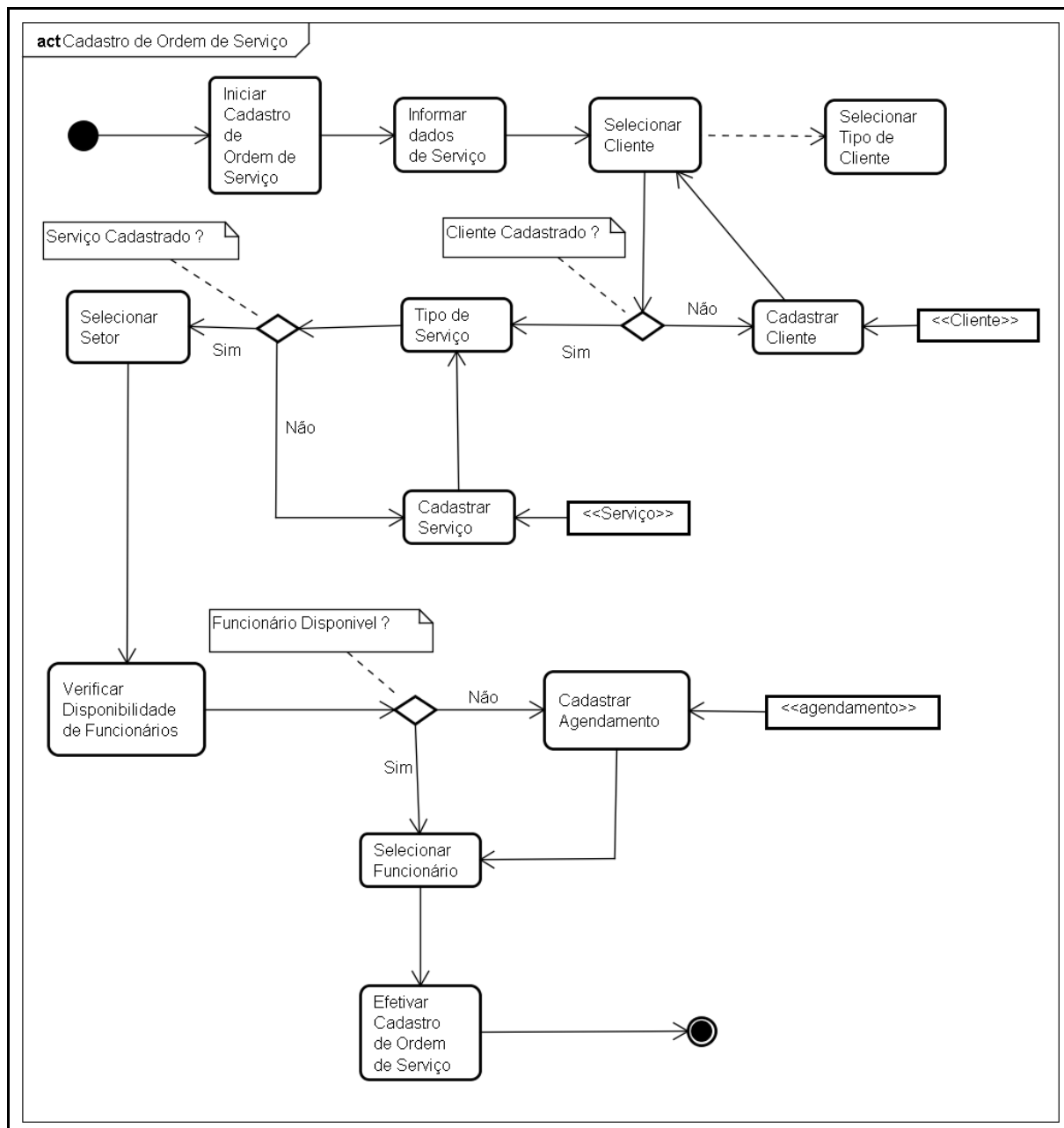


Figura 35 – Diagrama de Atividade Cadastrar Ordem de Serviço

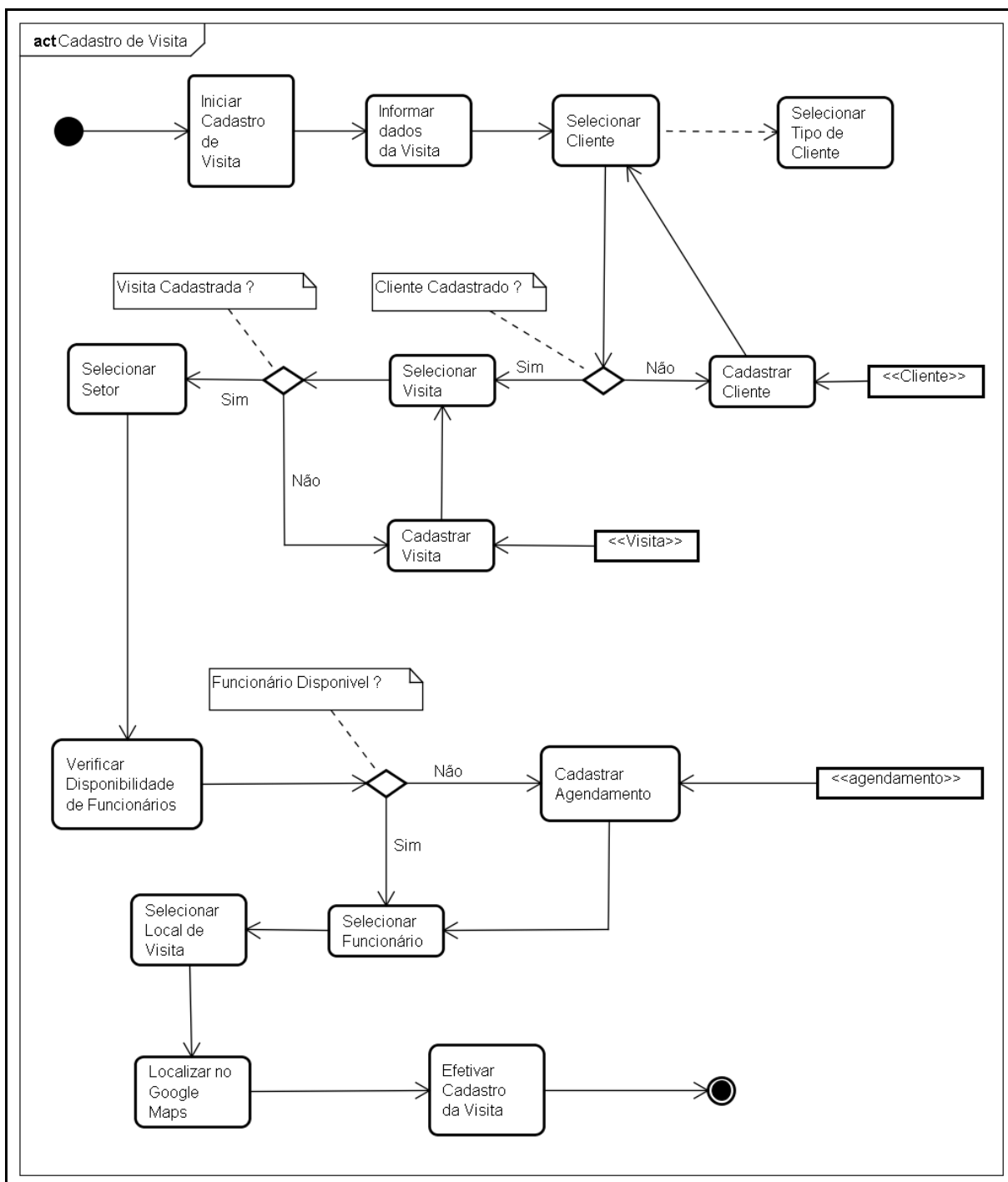


Figura 36 – Diagrama de Atividade Cadastrar Visita

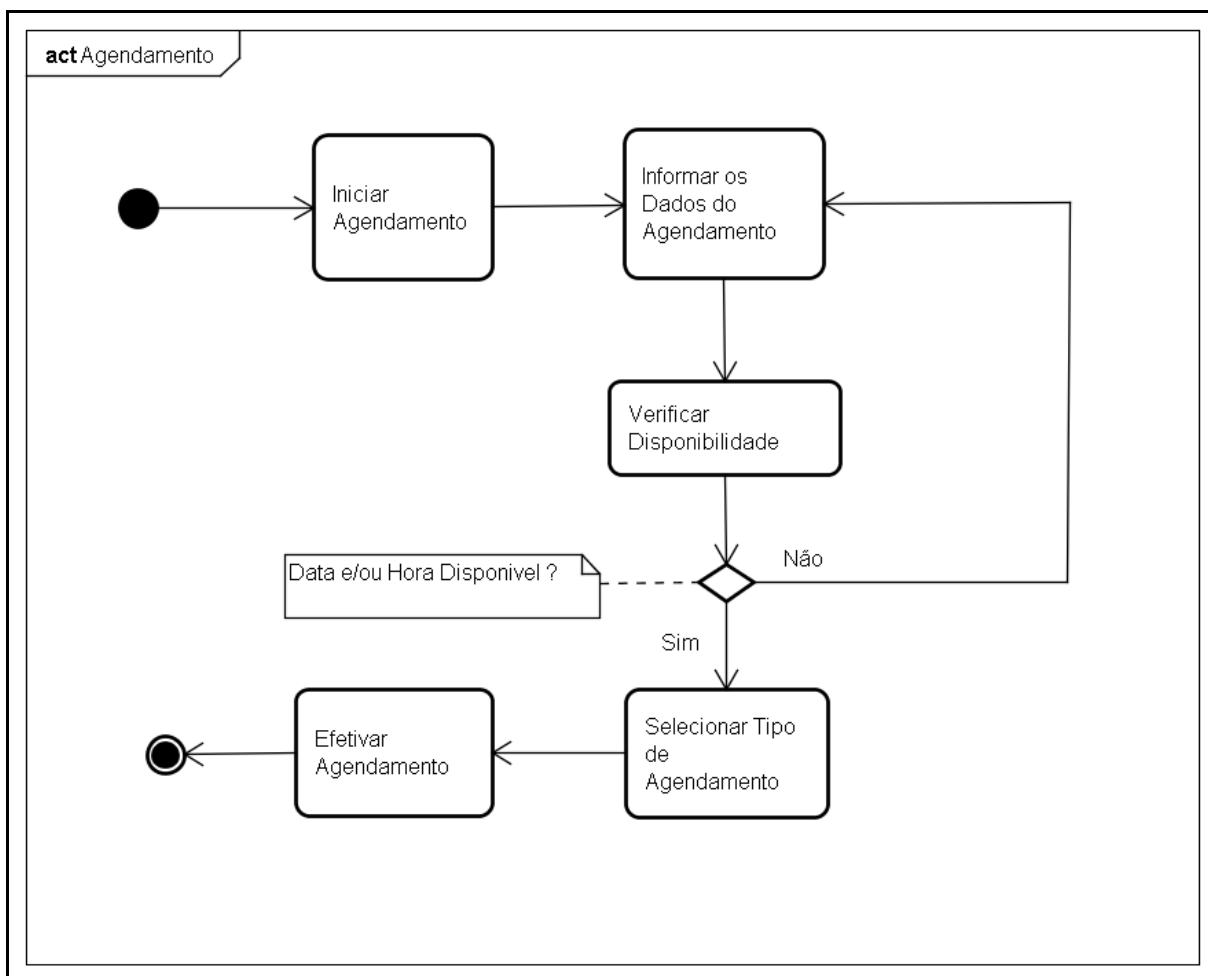


Figura 37 – Diagrama de Atividade Cadastrar Agendamento

Os diagramas de atividade apresentados ilustram o processo de uma determinada atividade.

3.5 – DIAGRAMA DE SEQUENCIAS

Um diagrama de sequência representa a troca de mensagens entre objetos de um determinado sistema. O que possibilita a visualização da troca de informação através de mensagens enviadas entre objetos contendo informações relacionadas ao resultado de uma determinada atividade (BOOCH; JACOBSON; RUMBAUCH, 2005).

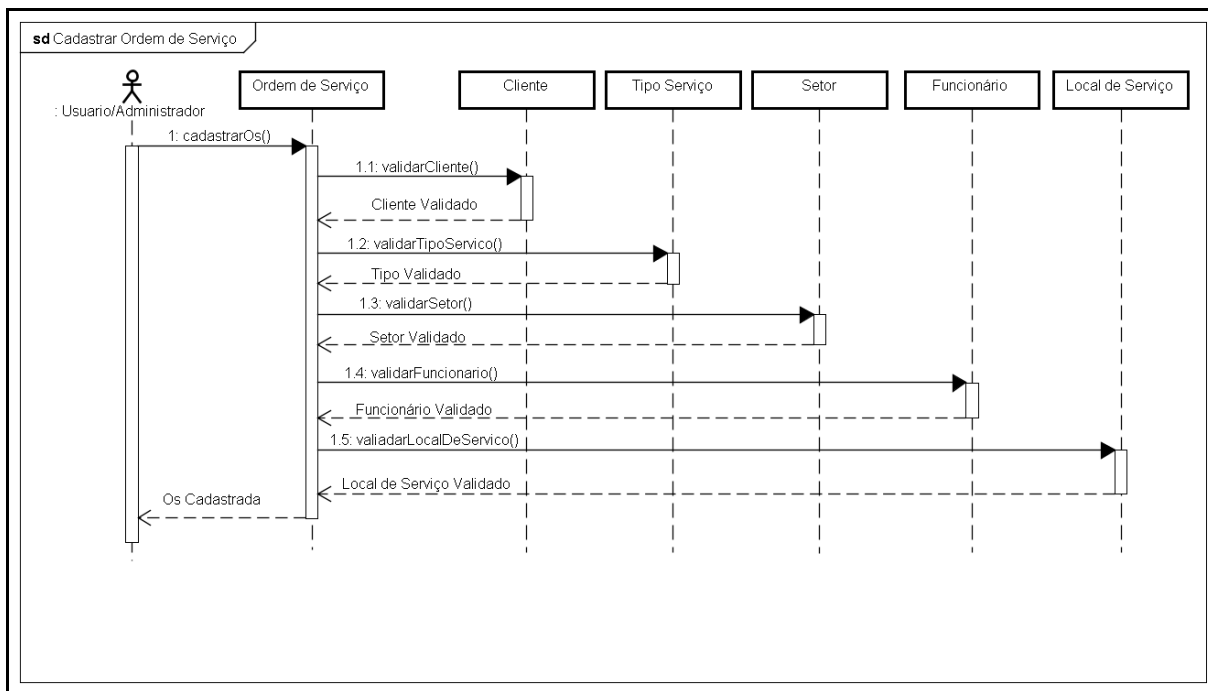


Figura 38 – Diagrama de Sequencia Cadastrar Ordem de Serviço

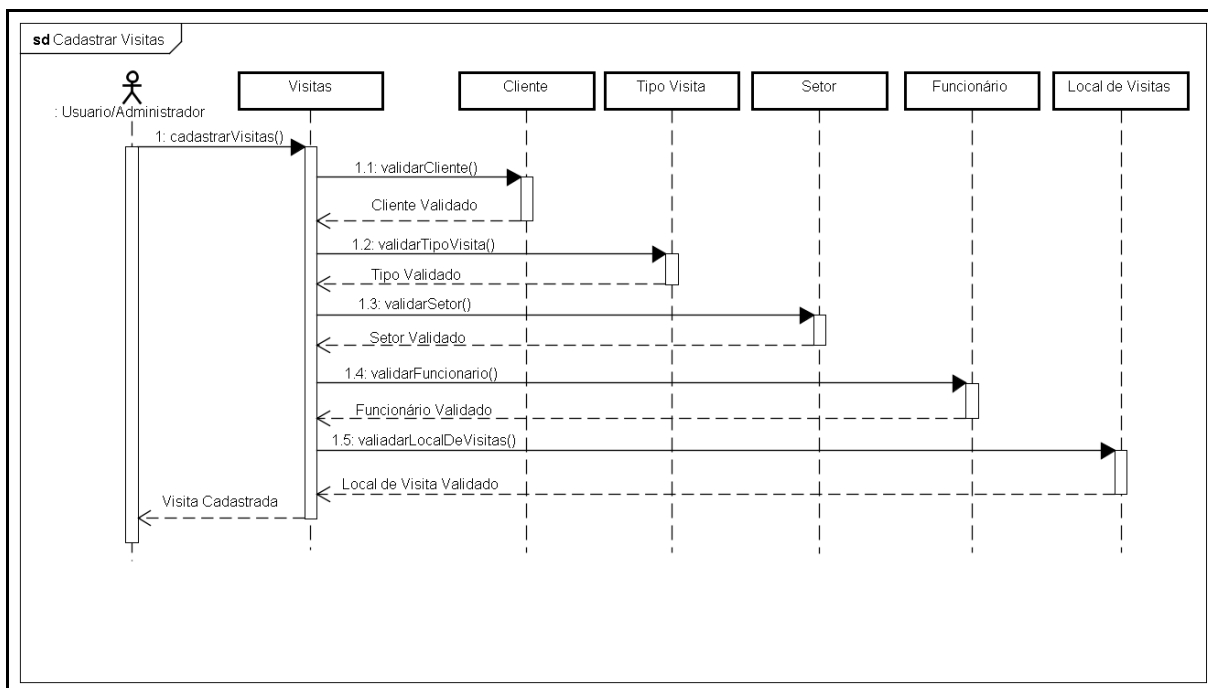


Figura 39 – Diagrama de Sequencia Cadastrar Visita

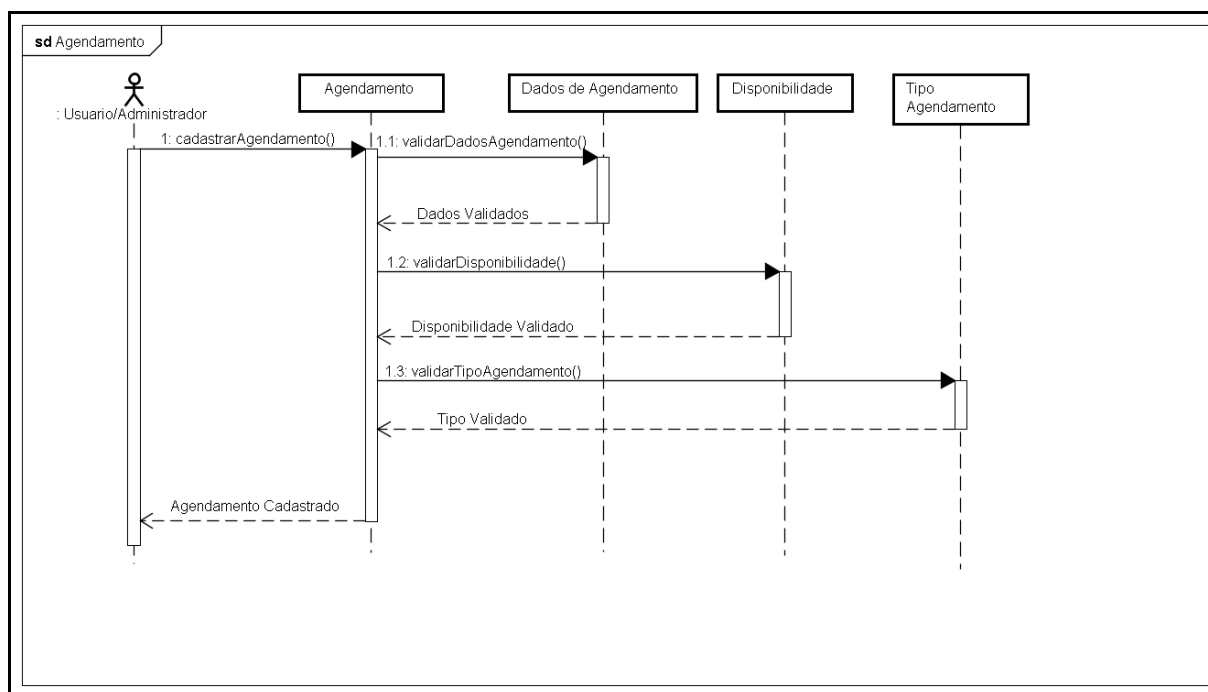


Figura 40 – Diagrama de Sequencia Cadastrar Agendamento

Os diagramas de sequencia apresentados ilustram a troca de mensagens para a execução de uma determinada atividade solicitada pela entidade.

3.6 – DIAGRAMA DE CLASSE

O diagrama de classe ilustra um conjunto de classes e seus relacionamentos de forma estática de um sistema, eles são importantes não só para documentação, mas também para o desenvolvimento do sistema contribuindo de forma impactante para a consolidação do escopo das funcionalidades do projeto, fornecendo-nos um alto nível de modelagem de esquemas (BOOCH; JACOBSON; RUMBAUCH, 2005).

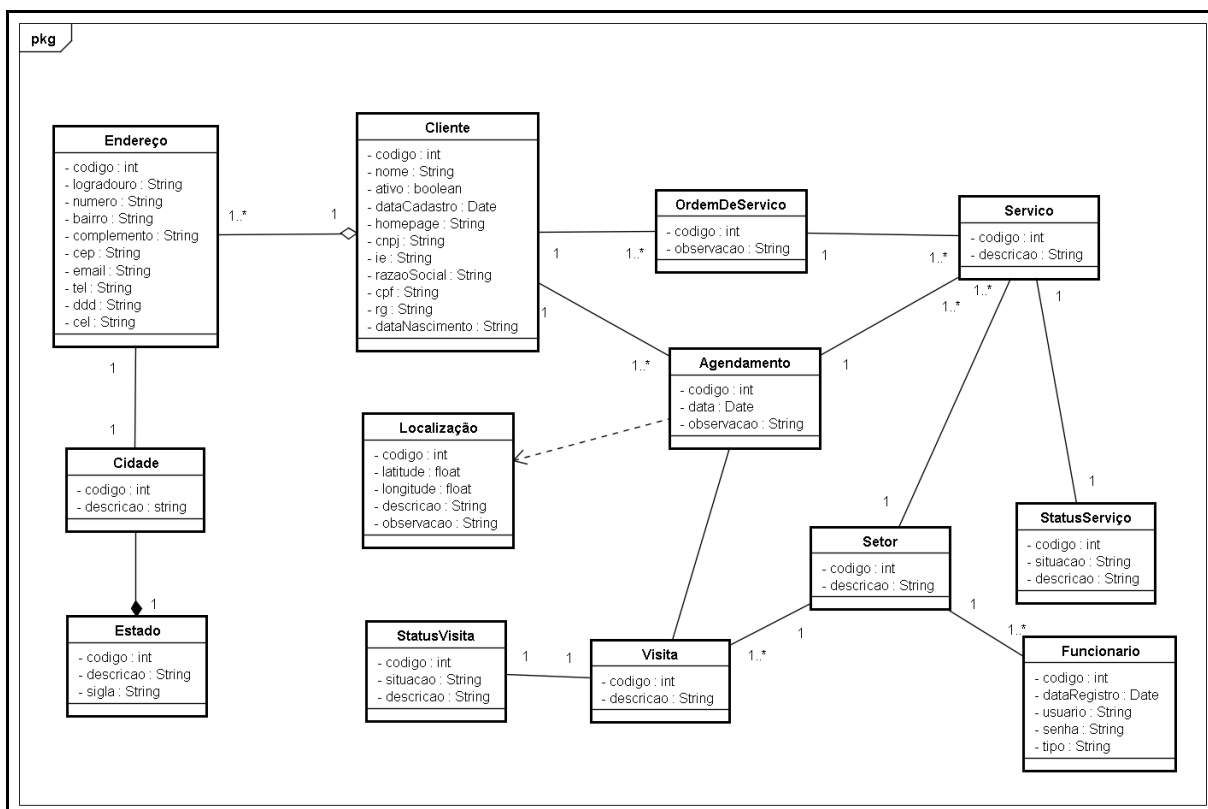


Figura 41 – Diagrama de Classe

O diagrama apresentado define todas as classes, atributos e relacionamentos existentes no sistema GOSV.

3.7 – MODELO ENTIDADE – RELACIONAMENTO

O Diagrama entidade-relacionamento ilustra a estrutura lógica do banco de dados descrito como um conjunto de objetos básicos chamados entidade e seus relacionamentos entre estes objetos, de forma a facilitar a especificação e desenvolvimento do esquema de banco de dados (REZENDE, 2005).

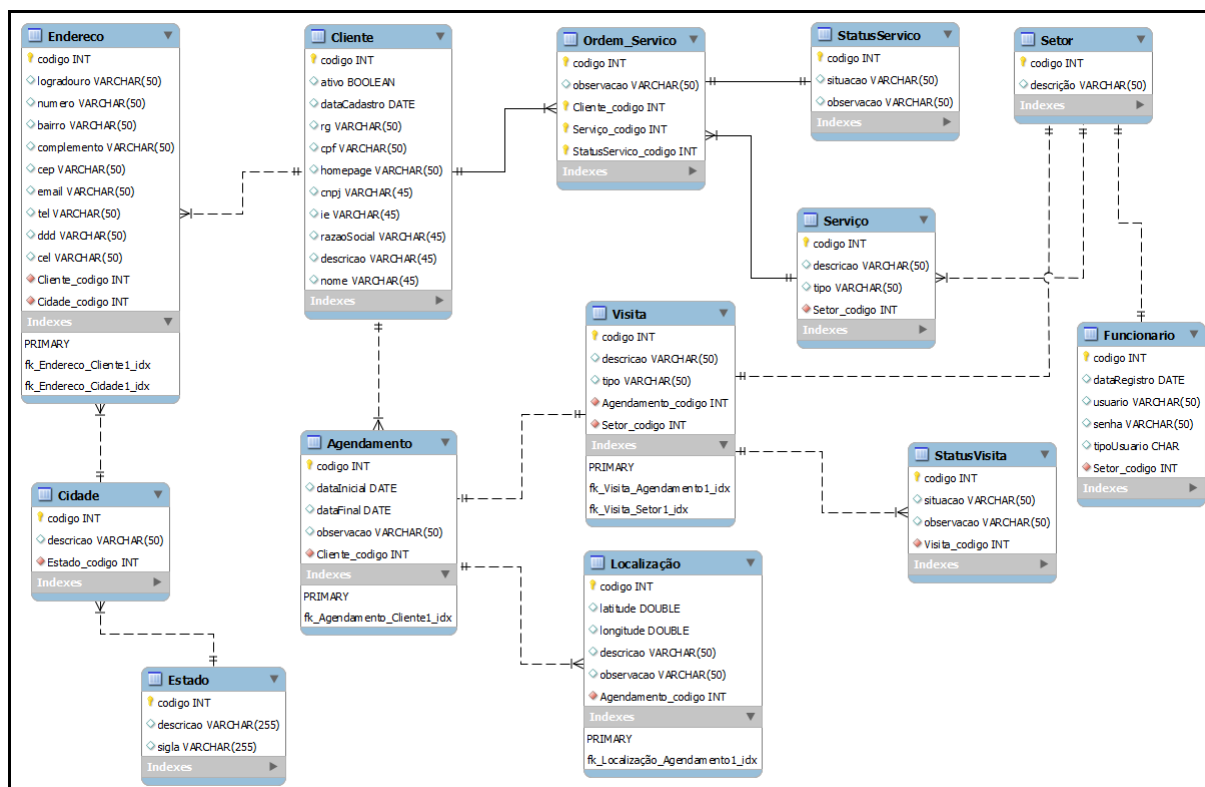


Figura 42 – Diagrama de Entidade Relacionamento

O diagrama apresentado ilustra as tabelas e relacionamentos existentes no banco de dados do sistema GOSV.

4 – ESTRUTURA DO PROJETO

Neste capítulo será apresentada a estrutura do projeto adotada para o desenvolvimento do trabalho de conclusão de curso, a qual consiste em fases e etapas. Para ilustrá-las, será apresentado o diagrama *WBS (Work Breakdown Structure)*, também conhecido como EAP Estrutura Analítica do Projeto, para complementar a estrutura do projeto os diagramas de sequenciamento de atividades para a determinação do tempo de cada atividade a ser desenvolvida e orçamento de equipamentos e serviços necessários para a conclusão do projeto.

4.1 – ESTRUTURA ANALITICA DO PROJETO

A EAP tem, como principal propósito, organizar a estrutura analítica do projeto, dividindo o trabalho em partes pequenas com alto nível de detalhamento do projeto com a finalidade facilitar o entendimento (PROJECT MANAGEMENT INSTITUTE, 2004).

As etapas do desenvolvimento da aplicação são ilustradas na Figura 40 que apresenta o diagrama EAP que será utilizado para organizar e orientar o trabalho a ser desenvolvido:

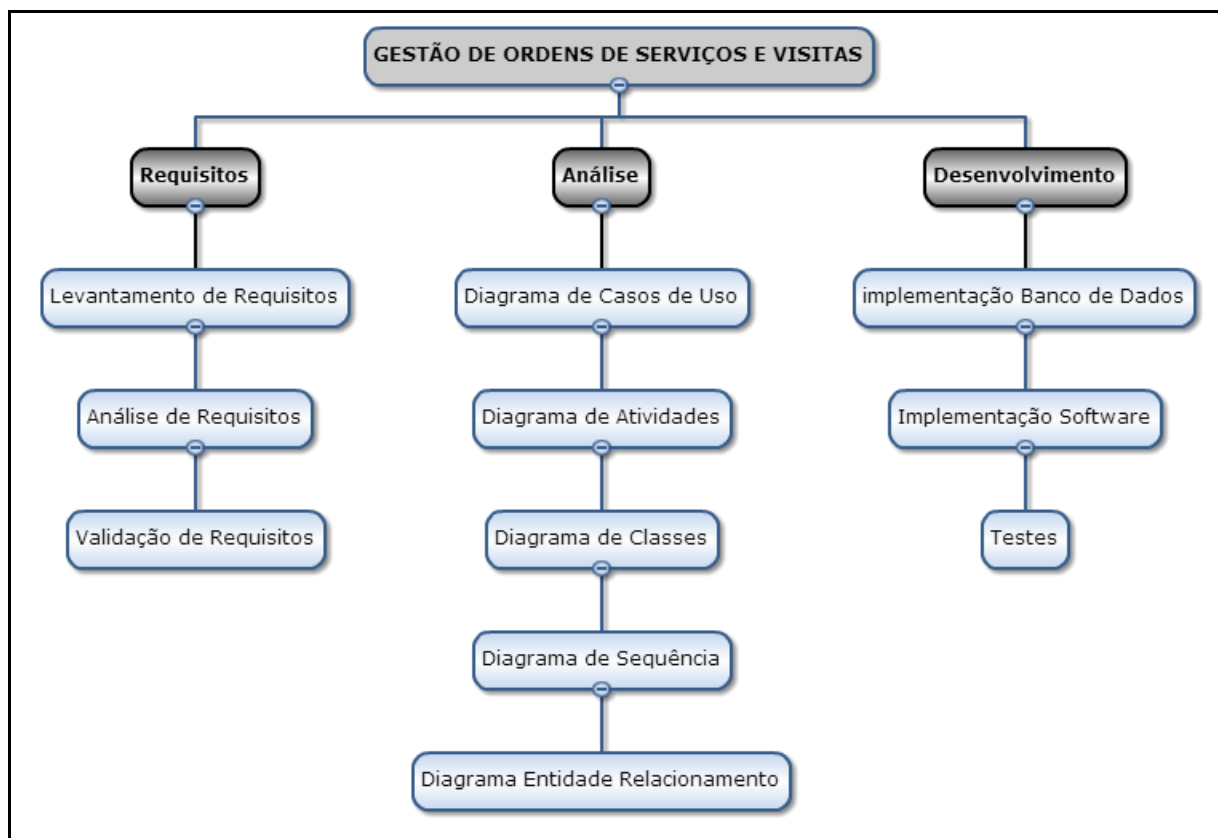


Figura 43 – Estrutura Analítica do Projeto

4.2 – SEQUENCIAMENTO DAS ATIVIDADES

O diagrama de sequenciamento de atividades ilustra o tempo de duração para a realização de cada atividade do projeto em desenvolvimento, tendo como objetivo definir de forma lógica de execução das tarefas.

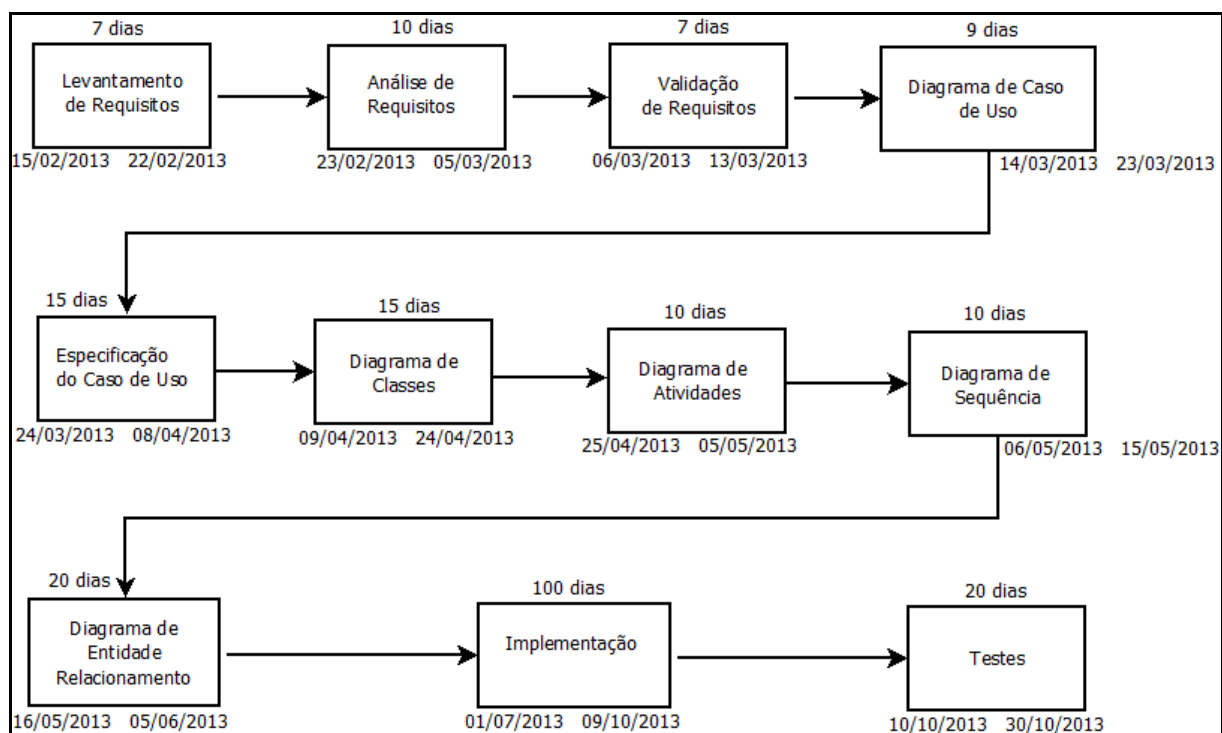


Figura 44 – Sequenciamento das Atividades

O diagrama apresentado ilustra o tempo necessário para conclusão de todas as atividades do trabalho de conclusão de curso.

4.3 – ORÇAMENTO

Os recursos necessários para análise e desenvolvimento do software GOSV são:

Recursos Físicos:

- 01 Analista de Sistema / Programador;

Analista	Quantidade Dias	Custo Dia	Total
Leonardo R Bastelli	103	R\$30,00	R\$3.090,00
Programador	Quantidade Dias	Custo Dia	Total
Leonardo R Bastelli	120	R\$25,00	R\$3.000,00
Custo Total			R\$6.090,00

Tabela 28 – Orçamento Analista e Programador

Equipamentos:

- 01 Notebook

Valor Unitário = R\$ 2.600,00

Depreciação (2 anos) = R\$2.600,00 / 24 = R\$108,34 /mês

Custo por dia = R\$108,34/ 26 (dias) = R\$4,16 (ao dia)

Custo do computador = R\$4,16 * 223 = R\$929,22.

- 01 Impressora jato de tinta

Valor = R\$300,00

Depreciação = R\$300,00 / 24 = R\$12,50

Custo do dia = R\$12,50 / 26(dias) = R\$0,49

Custo impressora = R\$0,49 *223 = R\$109,27.

Custo Total dos equipamentos R\$929,22 + R\$109,27 = R\$1.038,49

Recurso	Valor Total
Analista / Programador	R\$6.090,00
Equipamentos	R\$1.038,49
Valor Total do Projeto	R\$7.188,49

Tabela 29 – Orçamento Valor Total

5 – IMPLEMENTAÇÃO DO PROJETO

Para a implementação foi utilizado o ambiente de desenvolvimento eclipse com a linguagem de programação Java em conjunto com os *frameworks* *JSF*, *PrimeFaces* e *JPA*, no qual foi criado utilizando a separação por camadas, tornando o código mais organizado.

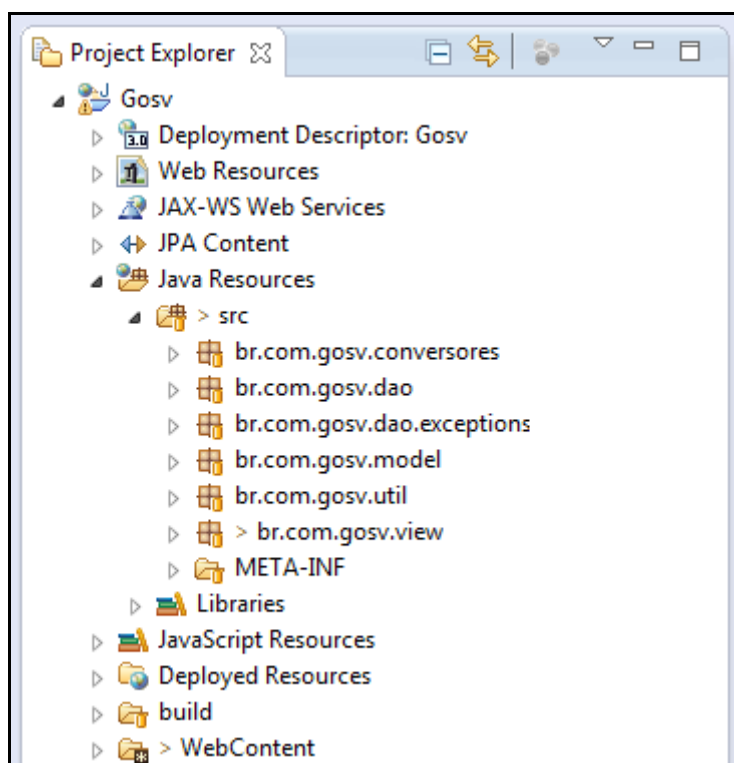


Figura 45 – Divisão de Camadas

A figura apresentada ilustra a divisão de camadas do projeto de trabalho de conclusão de curso.

5.1 – ORGANIZAÇÃO ESTRUTURAL DO PROJETO JAVA

O projeto GOSV apresenta sua organização dividida em seis pacotes: br.com.gosv.convertores, br.com.gosv.dao, br.com.gosv.dao.exceptions, br.com.gosv.model, br.com.gosv.util e br.com.gosv.view.

- **br.com.gosv.convertores:** pacote onde se localiza as classes de conversão de string para objeto e de objeto para string, atuando entre a página JSF e o Managed Bean, convertendo os textos da página automaticamente em objetos no Managed Bean.

```
1 package br.com.gosv.convertores;
2
3+ import javax.faces.component.UIComponent;
12
13 @FacesConverter (value="cliConverterAuto")
14 public class CliConverterAuto implements Converter{
15
16
17 @Override
18 public Object getAsObject(FacesContext fc, UIComponent uic, String string) {
19     if (string == null){
20         return null;
21     }
22     ClienteDao clienteDao = new ClienteDaoImp(JPAUtil.getEmf());
23     Cliente cliente = clienteDao.findCliente(Integer.parseInt(string));
24     return cliente;
25 }
26
27 @Override
28 public String getAsString(FacesContext fc, UIComponent uic, Object o) {
29     if (o== null){
30         return null;
31     }
32     Cliente cliente = (Cliente) o;
33     return cliente.getCodigo() + "";
34 }
35 }
```

Figura 46 – Classe Converter Cliente

A figura 46 ilustra a classe de conversão de cliente para uso de determinados componentes do *PrimeFaces* como *SelectOneMenu* e *AutoComplete*.

- **br.com.gosv.dao:** pacote onde se localiza as classes responsáveis pelo CRUD, onde sua instância é um objeto responsável por acessar os dados pela camada de persistência.

```
36
37  /* (non-Javadoc)
38   * @see br.com.Gosv.dao.UsuarioDao#create(br.com.Gosv.model.Usuario)
39   */
40  @Override
41  public void create(Usuario usuario) {
42      EntityManager em = null;
43      try {
44          em = getEntityManager();
45          em.getTransaction().begin();
46          em.persist(usuario);
47          em.getTransaction().commit();
48      } finally {
49          if (em != null) {
50              em.close();
51          }
52      }
53  }
```

Figura 47 – Classe UsuarioDao

A figura 47 ilustra a classe *UsuarioDao*, onde se implementa a criação de um novo usuário.

- **br.com.gosv.dao.exceptions:** pacote onde se localiza as classes responsáveis pelo tratamento de exceções da classe DAO.

- **br.com.gosv.model:** pacote onde se localiza as classes de mapeamento do objeto relacional, a classe representa uma determinada tabela do banco de dados.

```
20+ * To change this template, choose Tools | Templates
5 package br.com.gosv.model;
6
7+ import java.io.Serializable;
19
20- /**
21  *
22  * @author Leonardo
23  */
24 @Entity
25 @Table(name = "usuario")
26 @XmlElement
27 @NamedQueries({
28     @NamedQuery(name = "Usuario.findAll", query = "SELECT u FROM Usuario u")
29 })
29 public class Usuario implements Serializable {
30     private static final long serialVersionUID = 1L;
31-     @Id
32     @GeneratedValue(strategy = GenerationType.IDENTITY)
33     @Basic(optional = false)
34     @Column(name = "codigo")
35     private Integer codigo;
36-     @Basic(optional = false)
37     @Column(name = "nome")
38     private String nome;
39-     @Basic(optional = false)
40     @Column(name = "senha")
41     private String senha;
42-     @Column(name = "tipo_user")
43     private String tipoUser;
```

Figura 48 – Classe Usuário Model

A figura 48 ilustra a classe de mapeamento do objeto relacional, onde representa a tabela usuário do banco de dados.

- **br.com.gosv.util:** pacote onde se localiza a classe `JPAUtil`, responsável por instanciar um objeto `EntityManagerFactory` que prove todas as transações e serviços para a persistência.

```
1 package br.com.gosv.util;
2
3 import javax.persistence.EntityManager;
4
5
6
7 public class JPAUtil {
8     private static final EntityManagerFactory emf =
9         Persistence.createEntityManagerFactory("Gosv");
10
11     public static EntityManager getEntityManager() {
12         return emf.createEntityManager();
13     }
14
15     public static EntityManagerFactory getEmf() {
16         return emf;
17     }
18 }
19
```

Figura 49 – Classe `JPAUtil`

A figura 49 ilustra a classe `JPAUtil`, onde é criado um `EntityManagerFactory` para a unidade de persistência "Gosv", após a criação do `Factory` é criado uma `EntityManager` para que seja possível trabalhar com as entidades persistentes.

- **br.com.gosv.view:** pacote onde se localiza as classes *Managed Beans* responsáveis por receber os dados enviados pelo usuário através da tela, tratar as requisições e disponibilizar os resultados.

```

1  package br.com.gosv.view;
2
3  import java.io.Serializable;
21
22  @ManagedBean (name="clienteMB")
23  @ViewScoped
24  public class ClienteMB implements Serializable {
25      private static final long serialVersionUID = 1L;
26      private ClienteDao clienteDao;
27      private Cliente cliente;
28      private Cliente clienteEd;
29      private List<Cliente> clientes;
30      private int startingAt;
31      private int maxPerPage;
32      private Cliente selc;
33      //criar uma nova instancia do cliente para (managedBean)
34      public ClienteMB(){
35          this.clienteDao = new ClienteDaoImp(JPAUtil.getEmf());
36          this.cliente = new Cliente();
37          this.clienteEd = new Cliente();
38          this.clientes = null;
39      }
40      public String save(Cliente cliente){
41          FacesContext context = FacesContext.getCurrentInstance();
42          FacesMessage.Severity severity;
43          String message;
44          try {
45              Date dataAtual = new Date(System.currentTimeMillis());
46              cliente.setDataCadastro(dataAtual);
47              this.clienteDao.create(cliente);
48              severity = FacesMessage.SEVERITY_INFO;
49              this.cliente = new Cliente();
50              this.clientes = this.clienteDao.findClienteEntities();
51              message = " " + cliente.getNome() + " Salvo Com sucesso";
52          } catch (Exception e) {
53              message = cliente.getNome() + "Erro - valores incorretos" + e.getMessage();
54              severity = FacesMessage.SEVERITY_ERROR;
55          }
56          FacesMessage msg = new FacesMessage(severity, message, null);
57          context.addMessage(null, msg );
58          return "cliente?faces-redirect=true";
59      }
60

```

Figura 50 – Classe ClienteMB

A figura 50 ilustra o *Managed Bean* ClienteMB responsável por receber o objeto da página cliente.xhtml, executar a logica e devolver o resultado da solicitação do usuário.

Além dos pacotes de classes java apresentados anteriormente, também há um diretório importante chamado *WebContent*, onde é armazenado as páginas *JSF*, desenvolvidas em conjunto do *Framework PrimeFaces*.

A figura 51 ilustra o desenvolvimento de uma página *JSF* com a utilização de *tags* do *framework PrimeFaces*:

```

4<f:view xmlns="http://www.w3.org/1999/xhtml"
5  xmlns:ui="http://java.sun.com/jsf/facelets"
6  xmlns:h="http://java.sun.com/jsf/html"
7  xmlns:p="http://primefaces.org/ui"
8  xmlns:f="http://java.sun.com/jsf/core"
9  xmlns:pm="http://primefaces.org/mobile" contentType="text/html"
10 renderKitId="PRIMEFACES_MOBILE">
11
12<pm:page title="GESTÃO DE ORDEM DE SERVIÇO E VISITAS" mini="true">
13  <pm:view switch="d">
14    <pm:header title="GESTÃO DE ORDEM DE SERVIÇO E VISITAS" switch="b">
15      <f:facet name="right">
16        <h:form>
17          <p:commandButton process="@this" value="Home" icon="home"
18            action="#{navegarPaginaMB.home}" />
19        </h:form>
20      </f:facet>
21    <f:facet name="left">
22      <h:outputLink value="index.jsf?" process="@this">Sair</h:outputLink>
23    </f:facet>
24  </pm:header>
25  <pm:content>
26    <p:accordionPanel id="ac1" multiple="true" activeIndex="-1">
27      <p:tab title="Consulta de Ordem de Serviço">
28        <p:outputLabel value="Consulta de Ordem de Serviço pelo Código :"
29          style="font-weight:bold;font-size:17px;text-align:center;" />
30        <br />
31        <p:autoComplete id="atc" value="#{ordemservicoMB.ordemServico}"
32          completeMethod="#{ordemservicoMB.buscaPorCodigo}"
33          minQueryLength="1" var="ordem" process="@this"
34          converter="osConverterAuto"
35          itemLabel="Codigo #{ordem.codigo} Equipamento #{ordem.modelo} #{ordem.fabricante}"
36          itemValue="#{ordem}" required="true" multiple="true">
37          <p:ajax event="itemSelect"
38            listener="#{ordemservicoMB.handleSelect}" update=":ac1:frm2 " />
39        </p:autoComplete>

```

Figura 51 – Página JSF consultaOrdemServico.xhtml

Por meio das *tags* `<pm:page>` e `<pm:view>` define-se o layout mobile disponível na biblioteca *PrimeFaces mobile*, logo tornando possível a visualização do mesmo software em diversos equipamentos que disponibilizem algum navegador.

5.2 – INTERFACES DA APLICAÇÃO

Ao acessar o sistema abrirá a página de *login*, contendo os campos para usuário e senha, para que o usuário seja validado no sistema.

GESTÃO DE ORDEM DE SERVIÇO E VISITAS

Logim

Usuario :

Senha :

Logar

WebApp Gestão de Ordem de Serviços e Visitas
Leonardo R. Bastelli

Figura 52 – Página inicial sistema GOSV

A figura 52 ilustra a página inicial do sistema, para que o usuário se valide no sistema, após a validação o usuário é redirecionado para a página de menu do sistema.

Menu

GESTÃO DE ORDEM DE SERVIÇO E VISITAS

CLIENTES

- Cadastro
- Consulta e Alteração

ORDEM DE SERVIÇO

- Cadastro
- Consulta e Alteração

AGENDAMENTO DE VISITAS

- Cadastro
- Consulta e Alteração

SERVIÇOS

- Cadastro/Consulta

SETOR

- Cadastro/Consulta

FUNCIONARIO

- Cadastro/Consulta

SISTEMA DE GESTÃO DE ORDEM DE SERVIÇO E VISITAS

WebApp Gestão de Ordem de Serviços e Visitas
Leonardo R. Bastelli

Figura 53 – Página menu sistema GOSV

A figura 54 ilustra o cadastro de cliente, onde se deve selecionar o tipo de cliente para que seja possível a visualização de determinados campos disponíveis para “Pessoa Física” e “Pessoa Jurídica”, estes campos são documentos necessários para cada tipo de cliente, tais como RG, CPF, CNPJ, Razão Social e Inscrição Estadual.

DADOS DO CLIENTE

Nome :

Nome

Pessoa Jurídica Pessoa Física

Data Nascimento :

Data Nascimento Ex: 99/99/9999

Telefone :

Telefone Ex: (99)9999-9999

Telefone Celular :

Celular (99)99999-9999

Email :

Email

Descrição :

Descrição

✓ Cadastro de Endereço

✓ Salvar

WebApp Gestão de Ordem de Serviços e Visitas
Leonardo R. Bastelli

Figura 54 – Página cadastro de cliente sistema GOSV

A figura 55 ilustra o cadastro de cliente, em um dispositivo móvel onde se deve efetuar o mesmo procedimento citado acima.



Figura 55 – Página cadastro de cliente (mobile) sistema GOSV

A figura 56 ilustra a consulta de cliente, a consulta do cliente esta disponível apenas pelo nome do cliente, utilizando um componente chamado "AutoComplete" disponível no *Framework PrimeFaces Mobile*, torna a consulta mais ágil devido ao método de consulta que a cada letra inserida é disparado um *select* retornando todos os clientes com a letra digitada, logo não a necessidade de um botão para efetuar a busca.



Figura 56 – Página consulta de cliente sistema GOSV

A figura 57 ilustra a consulta de cliente, em um dispositivo móvel onde se deve efetuar o mesmo procedimento citado acima.



Figura 57 – Página consulta de cliente (mobile) sistema GOSV

As figuras apresentadas acima ilustram a possibilidade de efetuar o mesmo processamento de informação tanto e um dispositivo móvel como em navegador padrão de um computador ou notebook.

6 – CONCLUSÃO

A falta de sistemas de informação em um ambiente corporativo agrega um déficit em seus processos, devido à emissão de ordens de serviços em fichas de papéis, além do gasto e desordem no armazenamento de documentos e da necessidade de um funcionário específico para essa função.

A proposta do trabalho foi a de desenvolver um sistema de gestão de ordem de serviços e visitas para empresa VNGT, devido a forma de execução de serviços por meio de papéis seguindo o fluxo de que a ordem de serviço deve ser emitida por equipamento e, após a emissão, a mesma é enviada ao técnico responsável por realizar o orçamento assim como elaborar o diagnóstico. Após essa atividade técnica, a ficha retorna para o setor de manutenção objetivando formalizar o orçamento junto ao cliente e, com a devida aprovação. Com o objetivo de minimizar e até mesmo eliminar diversas dificuldades inerentes a organizações ainda não informatizadas ou que fazem uso de *softwares* que não atendem às necessidades específicas, foi utilizado frameworks como o JPA que é uma especificação para padronizar o acesso aos frameworks de mapeamento do objeto relacional onde as classes Java são anotadas para poderem ser persistidas, o JSF é um framework MVC para construção de interfaces de usuário baseada em componentes para aplicações web, em conjunto ao JSF entra o PrimeFaces uma biblioteca de componentes para criação de interfaces de usuário rica.

O produto GOSV apresenta diversas funcionalidades implementadas com o intuito de facilitar a prestação de serviços e visitas a clientes da empresa que hoje faz uso da aplicação, permitindo um melhor controle de diversas informações tais como tipos de serviços, clientes, funcionários, ordens de serviço e visitas além do armazenamento de históricos sobre os visitas e serviços realizados.

6.1 – TRABALHOS FUTUROS

Para trabalho futuro, pretende-se desenvolver métodos de segurança através de filtros e controle de sessão de acordo com as permissões de cada usuário, desenvolver um modulo de controle de estoque para as peças que são utilizadas em determinados equipamentos. Além disso, existe vontade de programar um método de “avise”, para que a cada nova ordem de serviço ou agendamento emitido para um determinado funcionário seja possível receber um aviso instantâneo de que foi aberto um chamado para este funcionário, assim agregando uma maior agilidade no lado do funcionário.

REFERÊNCIAS

BAZZOTTI, Cristiane; GARCIA, Elias. **A Importância do sistema de informação gerencial para tomada de decisões**. In: Seminário do centro de Ciências Sociais Aplicadas de Cascavel, 6, 2007, Cascavel, Brasil. Resumos. Cascavel: CCSA, 2007.

BOOCH, Grady; JACOBSON, Ivar; RUMBAUCH, James. **UML – Guia do Usuário**. 2ª Edição. Tradução Fábio Freitas da Silva e Cristiana de Amorim Machado. Rio de Janeiro: Elsevier, 2005.

CAELUM. **FJ11 Java e Orientação Objetos**. Disponível em <<http://www.caelum.com.br/apostilas/>>. Acesso em 20/06/2013.

CAELUM. **FJ21 Java para Desenvolvimento Web**. Disponível em <<http://www.caelum.com.br/apostilas/>>. Acesso em 20/06/2013.

ÇIVICI, Çağatay. **Prime Faces Mobile User's Guide**. Disponível em <<http://www.primefaces.org/documentation.html>>. Acesso em 28/05/2013.

ÇIVICI, Çağatay. **Prime Faces Mobile Guide**. Disponível em <<http://www.primefaces.org/documentation.html>>. Acesso em 29/06/2013.

COMPUTERWORLD. **Com o aumento do uso de smartphones e tablets, gestão de dispositivos vai decolar**. Estados Unidos. Disponível em <<http://computerworld.uol.com.br/negocios/2012/10/26/com-aumento-do-uso-de-smartphones-e-tablets-gestao-de-dispositivos-vai-decolar/>> Acesso em 11/03/2013.

DEVMEDIA. **Entenda como funciona a Java Virtual Machine (JVM)**. Disponível em <<http://www.devmedia.com.br/entenda-como-funciona-a-java-virtual-machine-jvm/27624>>. Acesso em 28/06/2013.

DEVMEDIA. **JPA 2.0 Persistência a toda Prova**. Disponível em <<http://www.devmedia.com.br/jpa-2-0-persistencia-a-toda-prova-java-magazine-81/17437>>. Acesso em 23/03/2013.

DEVMEDIA. **Introdução ao Java EE 6**. Disponível em <<http://www.devmedia.com.br/introducao-java-ee-6/21364>>. Acesso em 16/05/2013.

DEITEL, H.M; DEITEL, P.J. **Java – Como programar**. 8ª Edição. Tradução Edson Furmankiewicz. São Paulo: Person, Prentice Hall, 2010.

EXAME. **Brasil tem 88,2 milhões de acessos online em Janeiro**. São Paulo. Disponível em <<http://exame.abril.com.br/tecnologia/noticias/brasil-tem-88-2-milhoes-de-acessos-de-banda-larga-fixe-e-mov>>. Acesso em 10/03/ 2013.

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

GONÇALVES, Edson. **Desenvolvendo Aplicações Web com JSP, Servlet, JavaServer Faces, Hibernate, EJB 3 Persistence e Ajax**. Rio de Janeiro: Ciência Moderna, 2008.

HORSTMANN, Cay S; CORNEL, Gary. **Core Java Volume 1 Fundamentos**. 8ª Edição. Tradução Carlos Schafranski e Edson Furmankiewicz. São Paulo: Pearson Prentice Hall, 2010.

HORSTMANN, Cay S; DAVID, Geary. **Core JavaServer Faces**. 1ª Edição. Tradução Marcelo Soares. São Paulo: Alta Books, 2005.

JAVAFREE; Fórum. Disponível em: <<http://javafree.uol.com.br/wiki/Java>>. Acesso em 08/05/2013.

LOCKOW, Décio; MELO, Alexandre. **Programação Java para Web**. São Paulo: Novatec, 2010.

MYSQL. **MySQL Documentation**. Disponível em: <<http://dev.mysql.com/doc/index-archive.html>>. Acesso em 27/05/2013.

PRIMEFACES. **Why PrimeFaces**. Disponível em: <<http://www.primefaces.org/whyprimefaces.html>>. Acesso 29/06/2013.

PROJECT MANAGEMENT INSTITUTE. **PMBOK® – Um Guia do Conjunto de Conhecimentos em Gerenciamento de Projetos**. 3ª ed. Four Campus Boulevard, Newtown Square: Project Management Institute, 2004.

REZENDE, Denis Alcides. **Engenharia de software e sistemas de informação**. Rio de Janeiro: Brasport, 2005.

SANTOS, Anderson Carlos Bueno dos; VISOLE, Marcos Cezar; VACARI, Isaque; MEIRA, Carlos Alberto Alves; PIMENTA, Tiago Augusto; MAZZOTTI, Bruno Franciscon. **Desenvolvimento web com PrimeFaces: uso no projeto Banco de Dados Pragas Quarentenárias.**

<<http://ainfo.cnptia.embrapa.br/digital/bitstream/item/23892/1/p052.pdf>> acesso em 30/06/2013.

SILVEIRA, Paulo; SILVEIRA, Guilherme; LOPES, Sérgio; MOREIRA, Guilherme; STEPPAT, Nico; KUNG, Fabio. **Introdução à arquitetura e design de software.** Rio de Janeiro: Elsevier, 2012.

SOUSA, Lucilene. **Redesign da Informação no Processamento de Imagem.** Tese (Doutorado) – Universidade Federal de Santa Catarina, Florianópolis, 2004.

TEMPLE, André; MELLO, Rodrigo Fernandes; CALEGARI, Danival Taffarel; SCHIEZARO, Mauricio. **JSP, Servlets e J2EE.**

Disponível em <<http://www.inf.ufsc.br/~bosco/downloads/livro-jsp-servlets-j2ee.pdf>>. Acesso em 12/03/2013.

WAGNER, Rodrigo; MÄHLMANN, Luiz. **Sistema de Gestão de Produção de Eventos.** In: Seminário Intermunicipal de Pesquisa, 10, 2012, Guaíba, Brasil. Resumos. Guaíba, SIP, 2012.