

**SASSWEB — SISTEMA DE ACOMPANHAMENTO DE  
SOLICITAÇÕES DE SERVIÇO VIA WEB**

**AROLDO ESTEVES LIMA FILHO – MATRÍCULA 001022494**

**CARLA FREITAS CARNEVALE – MATRÍCULA 051020104**

**MARCIA PENHA DA SILVA MACHADO – MATRÍCULA 041010822**

**Centro Universitário da Cidade do Rio de Janeiro**

**Escola de Ciências Exatas e Tecnologia**

**Rio de Janeiro**

**Dezembro / 2008**

**SASSWEB — SISTEMA DE ACOMPANHAMENTO DE  
SOLICITAÇÕES DE SERVIÇO VIA WEB**

**AROLDO ESTEVES LIMA FILHO – MATRÍCULA 001022494**

**CARLA FREITAS CARNEVALE – MATRÍCULA 051020104**

**MARCIA PENHA DA SILVA MACHADO – MATRÍCULA 041010822**

Projeto Final apresentado como exigência da disciplina de Trabalho de Conclusão de Curso, sendo requisito para a conclusão do curso de Tecnologia em Análise e Desenvolvimento de Sistemas na UniverCidade.

**ALINE SILVESTRE ROSA**

**Orientadora**

**Centro Universitário da Cidade do Rio de Janeiro**

**Escola de Ciências Exatas e Tecnologia**

**Rio de Janeiro**

**Dezembro / 2008**

Este trabalho reflete a opinião dos autores e  
não necessariamente a da UniverCidade.

---

Aroldo Esteves Lima Filho

---

Carla Freitas Carnevale

---

Marcia Penha da Silva Machado

Agradecemos aos nossos pais pela formação e apoio que nos proporcionaram. Aos nossos professores pela indicação do caminho a trilhar. À nossa orientadora por sua ajuda e conselhos e aos membros da banca pela boa vontade e disponibilidade.

Os Autores

## SUMÁRIO

1. Introdução .....	11
1.1. Descrição do Tema .....	
1.2. Justificativa .....	12
1.3. Objetivos .....	
1.4. Análise do Ambiente Organizacional .....	
1.4.1. Identificação da Empresa .....	
1.4.2. Motivação .....	
1.4.3. Organograma da Empresa .....	13
1.4.4. Plataforma Tecnológica Necessária .....	
1.5. Levantamento do Sistema .....	14
1.5.1. Requisitos do Sistema .....	
1.5.2. Descrição das Regras de Negócio .....	15
1.5.3. Descrição do Mini Mundo .....	16
1.5.4. Detalhamento dos Processos de Negócio - Diagramas de Atividade .....	18
1.6. Metodologia de Desenvolvimento .....	21
1.6.1. Metodologia de Implantação .....	
1.7. Cronograma de Execução do Projeto .....	22
1.8. Orçamento do Trabalho .....	
2. Modelo de Casos de Uso .....	24
2.1. Diagrama de Casos de Uso .....	
2.2. Descrições Textuais dos Atores .....	
2.3. Descrições Textuais dos Casos de Uso .....	25
2.3.1. UC01 – Manter Cadastro de Funcionários .....	
2.3.2. UC02 – Manter Permissões .....	26
2.3.3. UC03 – Manter Cadastro de Áreas .....	27
2.3.4. UC04 – Manter Cadastro de Cargos .....	28
2.3.5. UC05 – Manter Cadastro de Funções .....	29

2.3.6. UC06 – Manter Coordenações .....	29
2.3.7. UC07 – Manter Cadastro de Serviços .....	30
2.3.8. UC08 – Alocar Executantes para o Serviço .....	31
2.3.9. UC09 – Validar as Solicitações e Gerar Programação de Serviços .....	32
2.3.10. UC10 – Encaminhar Serviços .....	
2.3.11. UC11 – Manter Cadastro de Solicitações .....	33
2.3.12. UC12 – Atender as Solicitações .....	
2.3.13. UC13 – Encerrar ou Cancelar as Solicitações .....	34
2.3.14. UC14 – Valorar a Satisfação com o Atendimento .....	
2.3.15. UC15 – Solicitar Detalhes .....	
2.3.16. UC16 – Fornecer Detalhes .....	35
2.3.17. UC17 – Gerar Relatório de Usuários .....	
2.3.18. UC18 – Gerar Relatório de Coordenadores .....	
2.3.19. UC19 – Gerar Relatório de Solicitações .....	36
2.3.20. UC20 – Gerar Relatório de Executantes .....	
2.3.21. UC21 – Gerar Relatório de Áreas .....	
2.3.22. UC22 – Gerar Relatório de Serviços .....	37
2.3.23. UC23 – Gerar Relatório de Atendimentos .....	
3. Modelo de Classes .....	38
3.1. Diagrama de Classes .....	
3.2. Dicionário das Classes .....	39
4. Modelo de Interações .....	41
4.1. Diagramas de Sequência .....	
5. Modelo de Transição de Estados .....	46
5.1. Funcionário .....	
5.2. Solicitação .....	
5.3. Executante .....	49
6. Projeto do Banco de Dados .....	50
6.1. Projeto Lógico do Banco de Dados .....	
6.2. Projeto Físico do Banco de Dados .....	51
6.3. Scripts das Tabelas do Banco de Dados .....	53
6.3.1. Tabela Areas .....	
6.3.2. Tabela Cargos .....	
6.3.3. Tabela Funcoes .....	
6.3.4. Tabela Permissoes .....	
6.3.5. Tabela Funcionarios .....	54
6.3.6. Tabela Usuarios .....	
6.3.7. Tabela Servicos .....	

6.3.8. Tabela Coordenacoes .....	55
6.3.9. Tabela Equipes	
6.3.10. Tabela Solicitacoes	
6.3.11. Tabela Detalhes .....	56
6.3.12. Tabela Programacao	
6.3.13. Tabelas Auxiliares	
6.4. Scripts das Views do Banco de Dados .....	57
6.4.1. Consulta ListaFuncionarios	
6.4.2. Consulta ListaServicos	
6.4.3. Consulta ListaCoordenadores	
6.4.4. Consulta ListaCoordenacoes .....	58
6.4.5. Consulta ConsultaExecutantes	
6.4.6. Consulta ListaEquipes	
6.4.7. Consulta ListaExecutantes	
6.4.8. Consulta ListaUsuarios .....	59
6.4.9. Consulta ListaAreas	
6.4.10. Consulta ConsultaSolicitacoes	
6.4.11. Consulta ConsultaDetalhes	
6.2.12. Consulta AreaCoordenador .....	60
6.4.13. Consulta ConsultaProgramacao	
6.4.14. Consulta ContaAtendimentos	
6.4.15. Consulta OrdemAtendimento .....	61
6.4.16. Consulta ServicoEquipe	
6.4.17. Consulta ListaSolicitados	
6.4.18. Consulta ListaAtendimentos	
7. Projeto da Interface Gráfica .....	62
7.1. Hierarquia das Telas	
7.2. Mapa de Navegação .....	64
7.3. Padronização dos Elementos da Interface Gráfica .....	68
8. Projeto de Implementação .....	69
8.1. Política de Login e Senhas .....	74
8.1.1. Atribuição do Login e da Senha Inicial	
8.1.2. Sigilo, Recuperação e Alteração de Senha	
8.2. Política de Manutenção e Backup	
9. Manual do Usuário .....	75
10. Conclusões e Trabalhos Futuros .....	84
11. Bibliografia .....	85

## FIGURAS E TABELAS

1. Introdução	
1.4. Análise do Ambiente Organizacional	
1.4.3. Organograma da Empresa (Fig.)	13
1.5. Levantamento do Sistema	
1.5.1. Requisitos do Sistema (Tab.)	14
1.4.2. Descrição das Regras de Negócio (Tab.)	15
1.5.4. Detalhamento dos Processos de Negócio – Diagramas de Atividade	
1.5.4.1. Efetuar Logon no Sistema (Fig.)	18
1.5.4.2. Emitir a Solicitação de Serviço (Fig.)	
1.5.4.3. Atribuir Executante ao Serviço (Fig.)	
1.5.4.4. Encaminhar o Serviço (Fig.)	19
1.5.4.5. Encerrar o Serviço (Fig.)	
1.5.4.6. Encerrar a Solicitação (Fig.)	20
1.5.4.7. Gerar Log (Fig.)	
1.5.4.8. Fornecer Detalhes (Fig.)	
1.5.4.9. Enviar E-mail (Fig.)	21
1.7. Cronograma de Execução do Projeto (Tab.)	22
1.8. Orçamento do Trabalho	
1.8.1. Planilha de Atividades e Apropriação de Horas de Trabalho (Tab.)	23
2. Modelo de Casos de Uso	
2.1. Diagrama de Casos de Uso (Fig.)	24
3. Modelo de Classes	
3.1. Diagrama de Classes (Fig.)	38
3.2. Dicionário das Classes	
3.2.1. Area (Tab.)	39
3.2.2. Cargo (Tab.)	
3.2.3. Funcao (Tab.)	



3.2.4. Permissao (Tab.) .....	39
3.2.5. Funcionario (Tab.) .....	
3.2.6. Servico (Tab.) .....	40
3.2.7. Solicitacao (Tab.) .....	
3.2.8. Detalhe (Tab.) .....	
3.2.9. Programacao (Tab.) .....	
4. Modelo de Interações	
4.1. Diagramas de Sequência	
4.1.1. Efetuar Logon no Sistema (Fig.) .....	41
4.1.2. Emitir a Solicitação de Serviço (Fig.) .....	42
4.1.3. Atribuir o Executante ao Serviço (Fig.) .....	
4.1.4. Encaminhar o Serviço (Fig.) .....	43
4.1.5. Encerrar o Serviço (Fig.) .....	
4.1.6. Encerrar a Solicitação (Fig.) .....	44
4.1.7. Solicitar Detalhes (Fig.) .....	
4.1.8. Fornecer Detalhes (Fig.) .....	45
4.1.9. Gerar Log .....	
5. Modelo de Transição de Estados	
5.1. Funcionário	
5.1.1. Efetuar Logon no Sistema (Fig.) .....	46
5.2. Solicitação	
5.2.1. Emitir a Solicitação de Serviço (Fig.) .....	
5.2.2. Encaminhar o Serviço (Fig.) .....	47
5.2.3. Encerrar o Serviço (Fig.) .....	
5.2.4. Encerrar a Solicitação (Fig.) .....	48
5.2.5. Fornecer Detalhes (Fig.) .....	
5.3. Executante	
5.3.1. Atribuir Executante ao Serviço .....	49
6. Projeto do Banco de Dados	
6.1. Projeto Lógico do Banco de Dados (Fig.) .....	50
6.2. Projeto Físico do Banco de Dados	
6.2.1. Tabela Areas (Tab.) .....	51
6.2.2. Tabela Cargas (Tab.) .....	
6.2.3. Tabela Funcoes (Tab.) .....	
6.2.4. Tabela Permissoes (Tab.) .....	
6.2.5. Tabela Funcionarios (Tab.) .....	
6.2.6. Tabela Usuarios (Tab.) .....	
6.2.7. Tabela Servicos (Tab.) .....	

6.2.8. Tabela Coordenacoes (Tab.) .....	52
6.2.9. Tabela Equipes (Tab.) .....	
6.2.10. Tabela Solicitacoes (Tab.) .....	
6.2.11. Tabela Detalhes (Tab.) .....	53
6.2.12. Tabela Programacao (Tab.) .....	
7. Projeto da Interface Gráfica	
7.1. Hierarquia das Telas	
7.1.1. Tela Inicial e Telas Principais (Fig.) .....	61
7.1.2. Telas no Logon de Solicitante (Fig.) .....	
7.1.3. Telas no Logon de Executante (Fig.) .....	62
7.1.4. Telas no Logon de Coordenador (Fig.) .....	
7.1.5. Telas no Logon de Administrador (Fig.) .....	
7.2. Mapa de Navegação	
7.2.1. Logon de Administrador (Fig.) .....	63
7.2.2. Logon de Coordenador (Fig.) .....	64
7.2.3. Logon de Executante (Fig.) .....	65
7.2.4. Logon de Solicitante (Fig.) .....	66
7.3. Padronização dos Elementos da Interface Gráfica	
7.3.1. Layout da Interface Gráfica (Fig.) .....	67
7.3.2. Padronização dos Elementos da Interface (Fig.) .....	
8. Projeto de Implementação	
Arquitetura MVC do Sistema (Fig.) .....	69
Esquema da Implementação Command (Fig.) .....	
Frameset da Interface Explodido (Fig.) .....	70
9. Manual do Usuário	
Logo do sistema (Fig.) .....	75
Telas de Exemplo (Fig.) .....	

## **CAPÍTULO 1 – INTRODUÇÃO**

A CISCEA — Comissão de Implantação do Sistema de Controle do Espaço Aéreo, é uma Organização do Comando da Aeronáutica, subordinada ao DECEA — Departamento de Controle do Espaço Aéreo, criada com a finalidade de desenvolver concepções, estudos, especificações, tecnologia e/ou equipamentos, fazer aquisições, desapropriações, construções, instalações elétricas e atividades decorrentes e correlatas, necessárias à implantação de um sistema no âmbito do SISCEAB — Sistema de Controle do Espaço Aéreo Brasileiro.

Assim, a CISCEA foi responsável pela instalação dos Centros Integrados de Defesa Aérea e Controle de Tráfego Aérea (CINDACTA); Centros de Controle de Aproximação (APP); Sistemas Radar; Tratamento e visualização de dados Radar e sistemas de telecomunicações aeronáuticas, bem como é responsável pela manutenção desses sistemas, o que gera inúmeras solicitações de serviços.

Com o objetivo de otimizar o emprego de seus recursos e o acompanhamento das solicitações, desenvolveremos um sistema para gerir e acompanhar as solicitações de serviço, suas evoluções no tempo e a satisfação no atendimento prestado pela CISCEA.

### **1.1 - DESCRIÇÃO DO TEMA**

Neste trabalho levantaremos as condições necessárias para informatizar o acompanhamento das solicitações de serviço, cadastrá-las, atribuir-lhes executantes e acompanhar sua evolução de modo a otimizar sua gestão, no prazo e no custo, sem desprezar a qualidade, já que está prevista também a avaliação do atendimento pelo solicitante do serviço.

## **1.2 - JUSTIFICATIVA**

A necessidade desse projeto se justifica porque a empresa lida com um grande número de solicitações de serviços diversos, internos e de prestação de manutenção, na sua rotina diária, mas não realiza nenhum controle/acompanhamento formal efetivo de qualquer tipo dessas solicitações, gerando baixo rendimento no atendimento e eventuais atrasos na realização dos serviços de manutenção e *helpdesk*.

## **1.3 - OBJETIVOS**

A implantação de um Sistema de Acompanhamento de Solicitações de Serviços internos e externos, servido tanto na rede interna (para atendimentos de *helpdesk* e manutenção) como na Internet (funcionários em atendimento externo de manutenção) possibilitará maior eficiência e eficácia no atendimento das solicitações com um melhor aproveitamento dos executantes, acompanhamento fácil da situação dessas solicitações, a manutenção de um histórico estatístico de desempenho, além de viabilizar um atendimento de melhor qualidade.

## **1.4 - ANÁLISE DO AMBIENTE ORGANIZACIONAL**

### 1.4.1 - Identificação da Empresa

A CISCEA localiza-se na Av. General Justo, 160 - Centro - Rio de Janeiro - CEP. 20021-300, inscrita com o CNPJ 00394429/0133-50. O Presidente da CISCEA é o Brigadeiro do Ar Carlos Vuyk de Aquino e a empresa está subdividida em uma Vice-Presidência, quatro Assessorias e cinco Divisões com cinco Sessões em cada uma. Os cargos de Presidente, Vice-Presidente e Diretores de Divisão são ocupados por militares, enquanto os cargos de Gerente de Sessão são ocupados por civis.

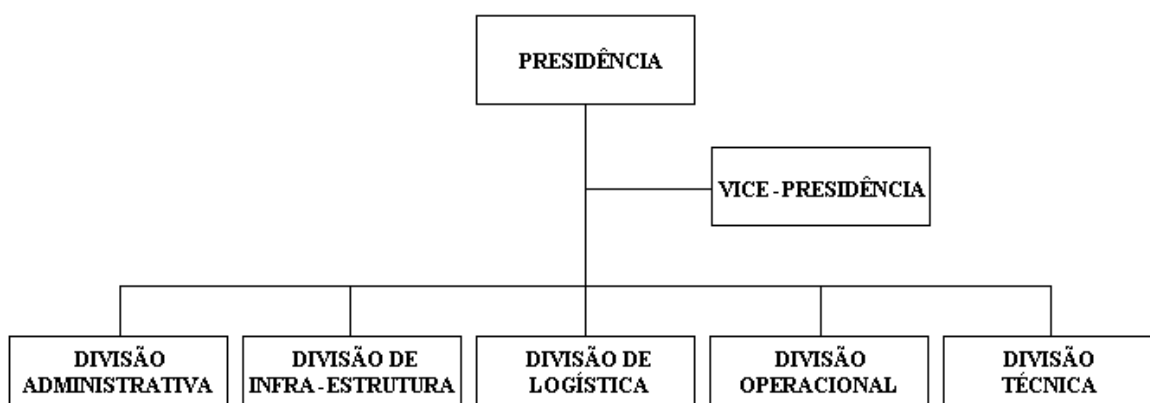
### 1.4.2 - Motivação

A idéia teve início com o vice-presidente anterior que notou a necessidade de uma ferramenta de controle e acompanhamento nos Centros de Apoio, Documentação e Informática. Os centros de Apoio e Documentação não faziam qualquer controle das solicitações, mas o de Informática já possuía um *helpdesk*. A aplicação imaginada deveria realizar a gestão, mas ser flexível, adequada a um ambiente de muito trabalho realizado por uma quantidade relativamente pequena de funcionários.

Conversando com os chefes desses centros, optou-se por implantar um sistema informatizado integrado para acompanhar as solicitações e execuções dos serviços.

De modo geral, há grande receptividade e expectativa sobre a implantação do sistema para acompanhamento das solicitações de serviço, tanto nas chefias como no quadro de funcionários.

#### 1.4.3 - Organograma da Empresa



#### 1.4.4 - Plataforma Tecnológica Necessária

O Sistema Operacional (SO) utilizado será o da empresa, Microsoft Windows 2000, mas nosso sistema deverá atender a outros SOs por sua característica de serviço via Internet. A empresa já possui um Servidor Apache e SGDB MySQL 5 para suporte aos seus Bancos de Dados (BD), além disso, todo o equipamento físico (*hardware*) necessário à implantação do sistema já está em uso na empresa.

Diante da estrutura que a empresa já dispõe e do que desejamos realizar, nossa escolha recai sobre o Java como linguagem de programação (portabilidade entre SOs e funcionalidade na Web) e MySQL 5 para o BD (uniformidade com o BD existente e aproveitamento do SGDB/DBA da CISCEA).

## 1.5 - LEVANTAMENTO DO SISTEMA

### 1.5.1 - Requisitos do Sistema

Requisitos	Descrição
R01	O sistema deverá controlar o acesso através de login e senha com permissões hierárquicas.
R02	O sistema deverá manter o cadastro de administradores do sistema.
R03	O sistema deverá manter o cadastro de coordenadores de área.
R04	O sistema deverá manter o cadastro de áreas de atuação (apoio, informática e documentação).
R05	O sistema deverá manter o cadastro de cargos e funções.
R06	O sistema deverá manter o cadastro de usuários, logins e permissões
R07	O sistema deverá manter o cadastro de solicitantes.
R08	O sistema deverá manter o cadastro de executantes dos serviços.
R09	O sistema deverá manter um cadastro de serviços, suas áreas, prazos padrão e prioridades.
R10	O sistema deverá manter um cadastro de solicitações de serviço.
R11	O sistema deverá permitir atribuição de executantes para os serviços.
R12	O sistema deverá permitir o acompanhamento do andamento das solicitações.
R13	O sistema deverá permitir o encerramento ou o cancelamento das solicitações.
R14	O sistema deverá permitir a alteração dos detalhes sobre o serviço nas solicitações emitidas e não encerradas nem canceladas.
R15	O sistema deverá permitir a valoração da satisfação com o atendimento.
R16	O sistema deverá respeitar a ordem das solicitações, prazo de execução e suas prioridades.
R17	O sistema deverá considerar dependências entre serviços.
R18	O sistema deverá gerar relatório de solicitantes (sob solicitação).
R19	O sistema deverá gerar relatório de coordenadores/áreas de atuação (sob solicitação).
R20	O sistema deverá gerar relatório de executantes com os filtros e critérios: livre, alocado, período e índice de aproveitamento (sob solicitação).
R21	O sistema deverá gerar relatório de serviços (sob solicitação).
R22	O sistema deverá gerar relatório de solicitações com os filtros e critérios: encerrada, em andamento, pendente, cancelada, por serviço, área e período (sob solicitação).
R23	O sistema deverá manter histórico dos atendimentos.
R24	O sistema deverá gerar relatório dos atendimentos (sob solicitação).
R25	O sistema deverá ter caráter de ferramenta, não de auditor, adequando-se ao modo de operação da empresa.

### 1.5.2 - Descrição das Regras de Negócio

RN	Descrição
RN01	Todo usuário é um funcionário da empresa com login cadastrado no sistema.
RN02	Hierarquias de acesso e permissões definidas em níveis conforme o cargo na empresa, além disso, no caso dos coordenadores, o login deverá considerar a área de atuação.
RN03	Cada nova solicitação de serviço é encaminhada ao coordenador.
RN04	Toda solicitação gerada possui solicitante, área, serviço e detalhe (informação adicional sobre o serviço e condições da execução).
RN05	O coordenador valida a solicitação, emitindo aviso para o solicitante, pelo sistema e por e-mail.
RN06	O coordenador aloca executantes disponíveis em equipes responsáveis por um serviço.
RN07	O coordenador encaminha a solicitação para uma equipe, mas o sistema é que seleciona o executante, considerando sua carga de trabalho.
RN08	As solicitações são atendidas na ordem de entrada, conforme o prazo padrão do serviço, a dependência de outro serviço e a prioridade (de 1 a 5, sendo 1 a maior prioridade).
RN09	O executante é que dá encerramento na execução do serviço.
RN10	O solicitante é que dá encerramento na solicitação executada, com valoração do atendimento (1 a 5, sendo 5 correspondente a ótimo).
RN11	Uma solicitação pendente ou em andamento não pode ser reemitida.
RN12	Uma solicitação pendente só admitirá alteração nos detalhes do serviço dessa solicitação.
RN13	Administrador tem visibilidade ilimitada, entretanto não pode alocar executantes nos serviços, encaminhar serviços ou encerrar/cancelar solicitações.
RN14	Coordenador tem ação apenas na sua área, mas pode consultar a programação dos serviços.
RN15	Solicitante tem visibilidade apenas das solicitações que emitiu.
RN16	Executante tem visibilidade apenas das solicitações que foram encaminhadas para sua equipe.
RN17	Toda operação gera uma entrada de log para efeito de histórico.
RN18	Administrador manterá o cadastro de Usuários, Permissões, Áreas, Cargos, Funções, Coordenações e Serviços.
RN19	Coordenador manterá o cadastro de Equipes de executantes dos serviços.
RN20	Um usuário poderá acumular papéis (permissões) no sistema, mas cada permissão será acessada isoladamente através de um login individual, exceto o papel de solicitante que estará embutido nos outros perfis.

### 1.5.3 - Descrição do Mini Mundo

A administração da CISCEA deseja informatizar a solicitação de serviços de manutenção e *helpdesk*, de modo a geri-las e acompanhá-las na execução dos serviços visando a obtenção de melhor desempenho e qualidade no atendimento. Esse sistema deverá ser servido na rede interna da empresa e na Internet. Os procedimentos desejados são descritos abaixo.

Dados dos funcionários, tais como matrícula, nome, área, cargo, função, e-mail e login, serão mantidos pelo sistema. Os dados referentes às Áreas, Cargos e Funções também serão mantidos pelo sistema, assim como deverá manter e controlar o acesso dos usuários.

O sistema será acessado apenas por funcionários da empresa — e todos os funcionários deverão acessá-lo —, através de login com usuário e senha, segundo uma hierarquia de permissões: Administrador, Coordenador de Área, Solicitante e Executante, conforme o cargo que esses funcionários ocupam. O Administrador tem permissão ampla, exceto para alocar Executantes nos serviços, encaminhar serviços, encerrar ou cancelar solicitações. Coordenadores estão limitados às suas Áreas, sendo consultada a Área do Coordenador no momento do login. Cada Solicitante acessa somente as solicitações que gerou e cada Executante as que foram encaminhadas para sua Equipe.

Um Solicitante emite uma solicitação de serviço contendo sua identificação, a Área e serviço desejados e uma descrição adicional sobre os detalhes ou condições do serviço. A solicitação emitida é validada pelo Coordenador da Área escolhida. Caso a solicitação não seja aceita, o Solicitante receberá um aviso pelo sistema e por e-mail. Se aceita, o Coordenador aloca o Executante disponível para a execução do serviço no prazo padrão definido para esse serviço, considerando sua prioridade, e emite aviso para o Executante pelo sistema e por e-mail. A solicitação passa ao estado de pendente, entrando na fila de espera. Ela poderá ser cancelada ou atendida. Quando a solicitação for atendida, o Executante encerrará o serviço e o Solicitante deverá encerrar a solicitação, valorando a sua satisfação com o atendimento prestado. Uma solicitação pendente ou em atendimento não poderá ser emitida novamente, mas deverá permitir a alteração no campo de informações adicionais sobre detalhes do serviço.

As datas e horas deverão ser obtidas pelo sistema diretamente do Sistema Operacional, sem qualquer intervenção dos usuários e, juntamente com a operação e a identificação dos usuários envolvidos, compor um log, um histórico das



solicitações, atendimentos e seus movimentos. A exclusão de dados cadastrais não será vedada, mas restrita, entretanto, não poderá haver exclusão no cadastro de solicitações.

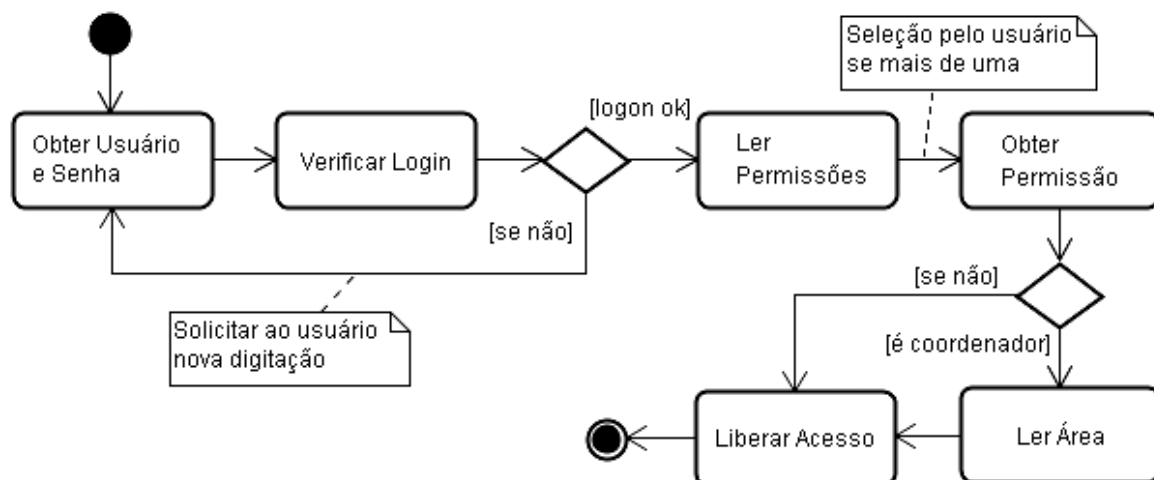
Deseja-se obter informações sobre os atendimentos, seus índices de aproveitamento e satisfação na execução. Por fim, o sistema deverá prever a emissão, a qualquer tempo e sob solicitação, de relatórios com critérios adequados, para visualização/impressão dos itens Área, Cargo, Função, Usuários, Serviços e Solicitações.

O sistema deverá ser adequado ao funcionamento da empresa também no aspecto da utilização do pessoal, possuindo o caráter de ferramenta e não de auditor. Assim, ele deverá possibilitar acúmulo de funções referentes às solicitações de serviço — um coordenador poderá coordenar mais de um serviço — e também de papéis — o coordenador de um serviço poderá ser executante de outro serviço. Entretanto, cada papel estará vinculado a um login, ou seja, durante a sessão de um determinado acesso, o usuário com permissões múltiplas, como coordenador e executante por exemplo, será apenas Coordenador ou apenas Executante.

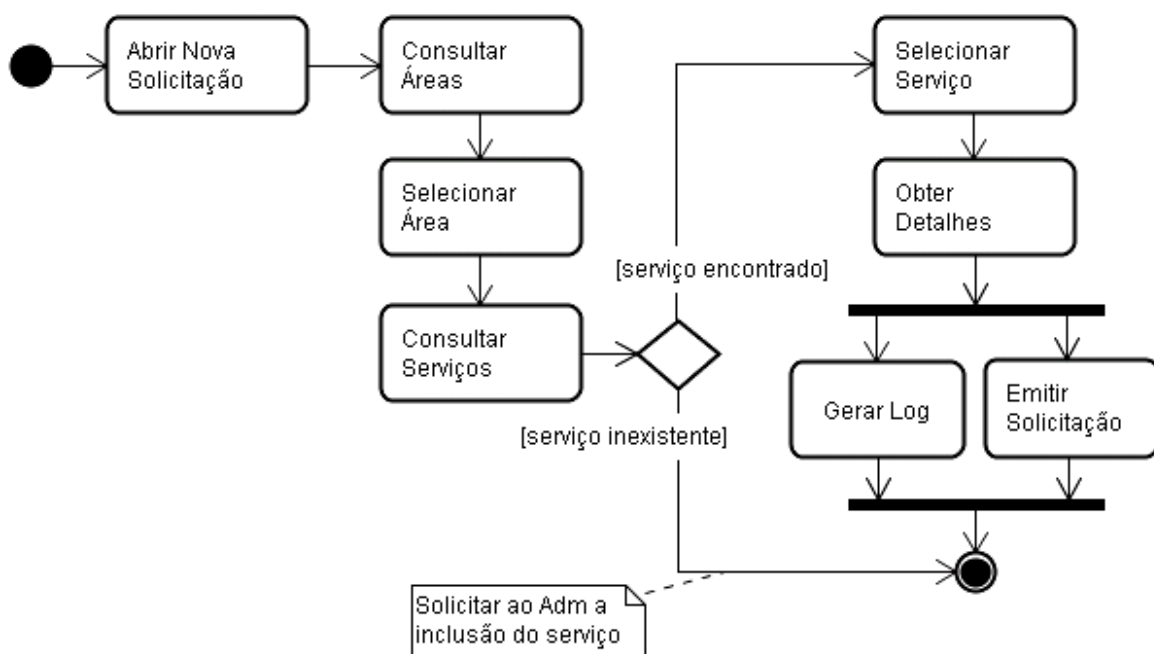
O único perfil do sistema presente em qualquer permissão de acesso é o de Solicitante, que terá suas funcionalidades sempre disponíveis para o usuário logado, durante a sua sessão.

## 1.5.4 - Detalhamento dos Processos de Negócio - Diagramas de Atividade.

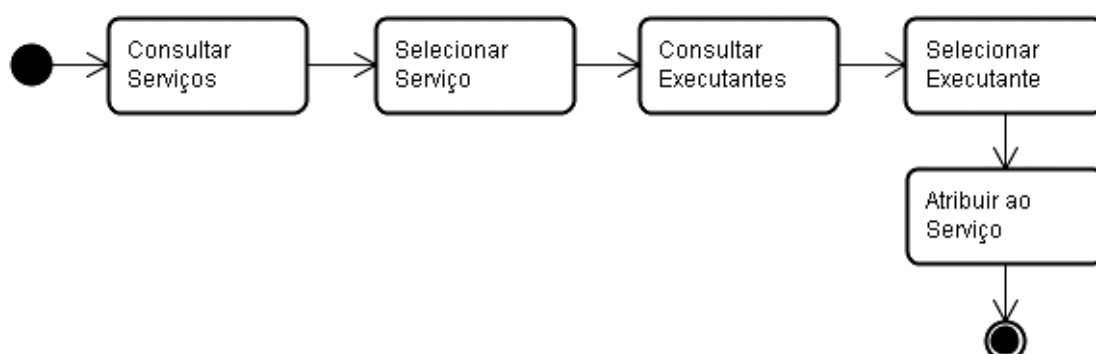
### 1.5.4.1 - Efetuar Logon no Sistema



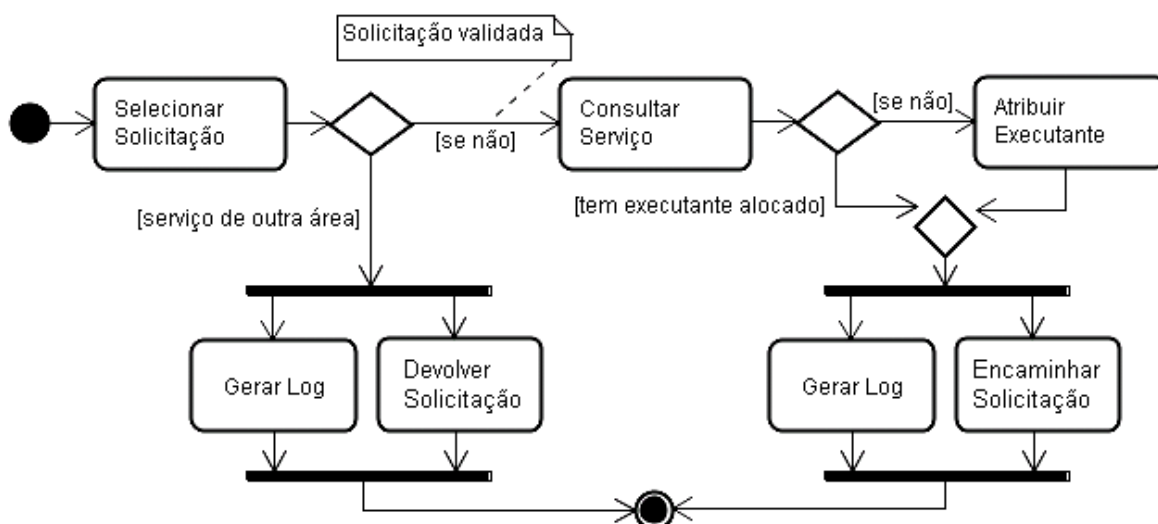
### 1.5.4.2 - Emitir a Solicitação de Serviço



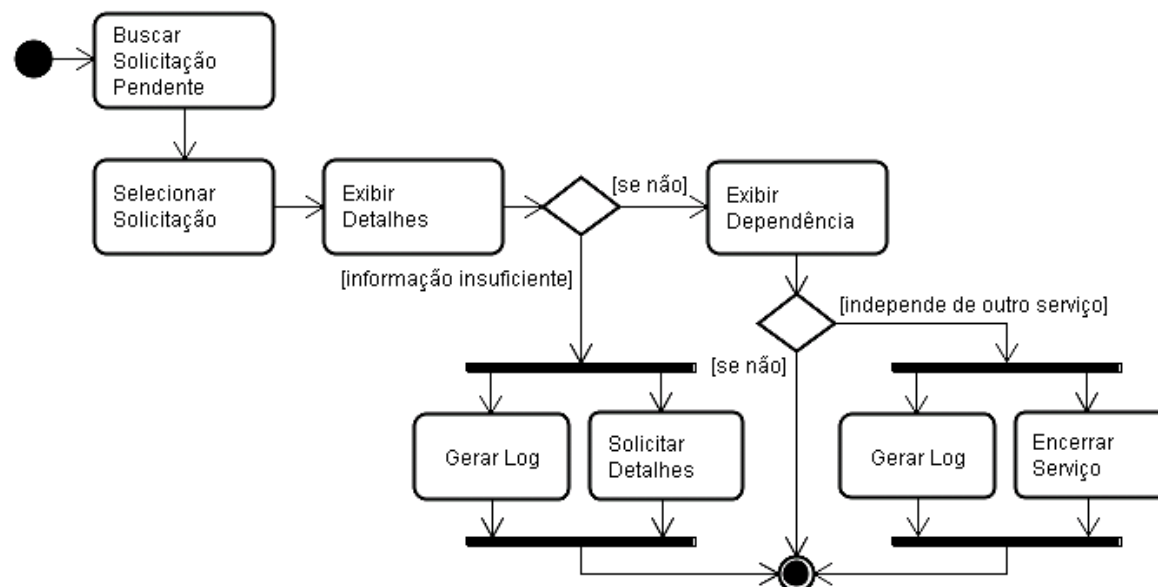
### 1.5.4.3 - Atribuir Executante ao Serviço



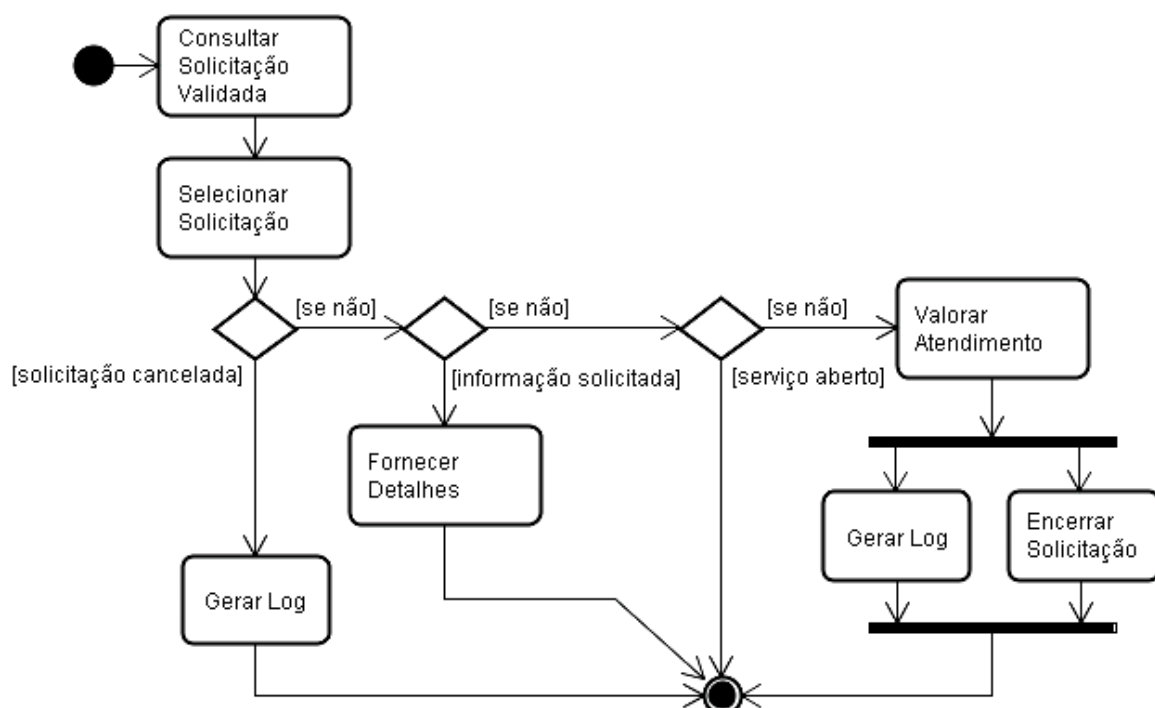
## 1.5.4.4 - Encaminhar o Serviço



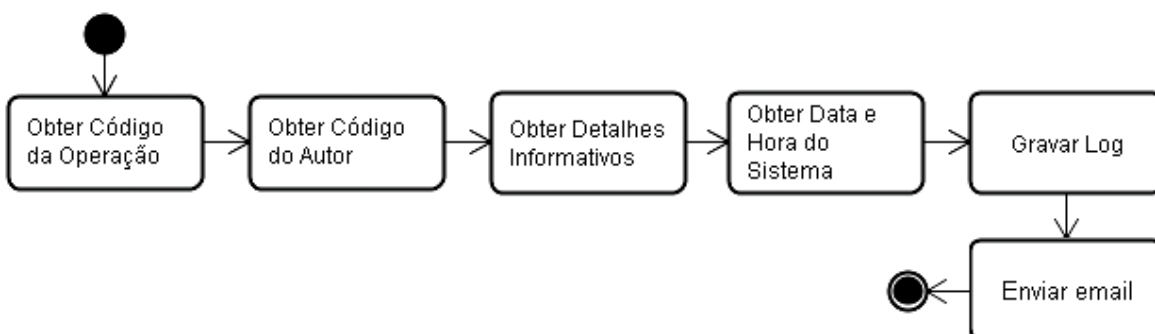
## 1.5.4.5 - Encerrar o Serviço



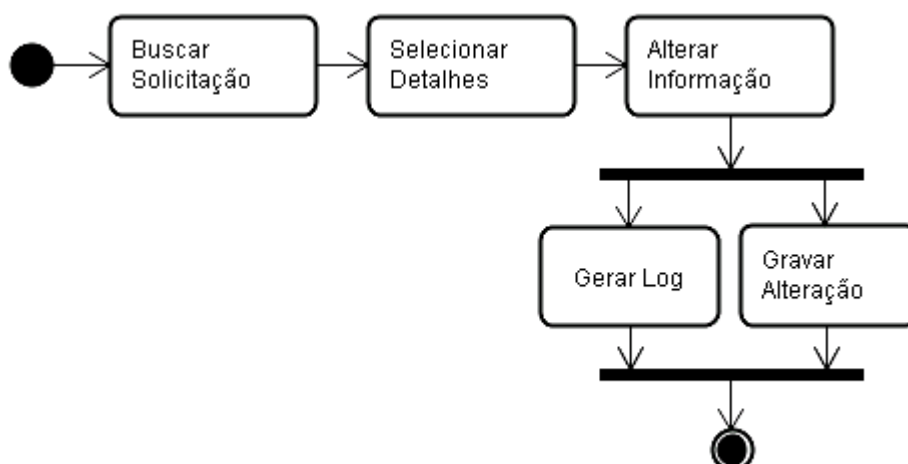
## 1.5.4.6 - Encerrar a Solicitação



## 1.5.4.7 – Gerar Log



## 1.5.4.8 – Fornecer Detalhes



#### 1.5.4.9 – Enviar E-mail



## 1.6 - METODOLOGIA DE DESENVOLVIMENTO

Faremos entrevistas com os funcionários envolvidos nas atividades, e na implantação, sobre os procedimentos de trabalho, coleta de informações para compor o BD do sistema e aquisição, junto ao setor responsável, das especificações de tabelas e campos referentes aos dados do BD de funcionários que serão utilizados pelo sistema de acompanhamento de solicitações de serviço, listaremos as necessidades e expectativas dos gerentes e demais funcionários.

É desejável o menor custo possível. Assim, optaremos por *software* livre. Utilizaremos Análise Orientada a Objetos e Padrão de Projeto MVC — *Model-View-Controller*, JSP — *Java Server Pages* e *Servlet* (Tecnologia para Web J2EE da Sun Microsystems) e como ferramenta de desenvolvimento o J2EE UML 5.3 Community. Faremos a codificação em Eclipse 3.2 WTP (*Web Tools Platform*) sobre um servidor Apache Tomcat 5.5 e banco de dados MySQL 5.

Oportunamente, faremos a seguinte observação: utilizamos o termo **usuário** para designar um funcionário genérico, de qualquer hierarquia, que acessa o sistema. As quatro hierarquias que definiremos, Administrador, Coordenador, Executante e Solicitante, possuem características próprias, entretanto, todas podem solicitar um serviço. Assim, Solicitante é um perfil existente em todas as hierarquias e, algumas vezes neste trabalho, utilizaremos o termo **usuário** com o significado de solicitante.

### 1.6.1 – Metodologia de Implantação

Em um momento anterior à implantação — e durante, junto com o treinamento para usuários —, faremos uma mobilização para formação cultural dos funcionários, através de palestras breves e distribuição de impressos informativos, visando a preparação e a educação dos usuários para a utilização do sistema (vide Anexo I).

Essa preparação será necessária, porque o sucesso e a eficácia do sistema que implantaremos, dependerá em grande parte, do modo como será utilizado, evitando as solicitações desnecessárias, os pedidos de pessoa para pessoa, a decisão da programação por parte dos executantes e a parcialidade na valoração dos atendimentos.

## 1.7 - CRONOGRAMA DE EXECUÇÃO DO PROJETO

Obs.: 4as-feiras assinaladas no topo do calendário.

	Ago			Set				Out					Nov				Dez			
	13	20	27	3	10	17	24	1	8	15	22	29	5	12	19	26	3	10	17	24
Apresentação das normas do TCC e roteiro / Montagem dos grupos e tema	.																			
Entrega do anteprojeto		.																		
Diagrama de Caso de Uso e descrição do caso de uso			.																	
Diagrama de Classe				.																
Diagrama de Sequência e Diagrama de Atividades					.															
Diagrama de Transição de Estados					.															
Entrega do Projeto Lógico					.															
Projeto Físico					.															
Projeto da Interface, padronização dos botões						.														
Layout das telas						.														
Projeto de Implantação, Manual do Usuário, Conclusões e Trabalhos Futuros																				
Apresentação dos projetos																24		12		
Revisão dos projetos																			13	19

## 1.8 - ORÇAMENTO DO TRABALHO

Considerando que não teremos custo de *hardware* e nem de *software*, o orçamento se resumirá ao cálculo das horas de trabalho dos profissionais envolvidos. O cálculo será feito sobre a planilha anexa contendo as atividades relativas ao desenvolvimento com uma duração total estimada em 45 dias úteis. Essa planilha será a justificativa para o cálculo das horas dos profissionais:

- ♦ Analista: R\$ 13,00 p/h total de 190h → R\$ 2.470,00
- ♦ Programador 2 (sr): R\$ 12,00 p/h total de 53h → R\$ 636,00
- ♦ Programador 1 (jr): R\$ 10,00 p/h total de 212h → R\$ 2.120,00 total 1: R\$ 5.226,00
- ♦ Gasto com material de apoio:
  - Toner para impressora: R\$ 150,00 (1 cartucho)
  - Papel A4 Sulfite 75g: R\$ 12,00 (500 folhas) x 2
  - Encadernação do Manual: R\$ 20,00

total 2: R\$ 194,00

**Orçamento:** 5.226,00 + 194,00 = 5.420,00 + fundo de reserva 1.000,00 = **R\$ 6.420,00**

### 1.8.1 – Planilha de Atividades e Apropriação de Horas de Trabalho

A planilha apresentada abaixo justifica o cálculo do custo total com os profissionais para o desenvolvimento e implantação do sistema, executado conforme descritos na Sequência de Atividades e Alocação de Recursos Humanos.

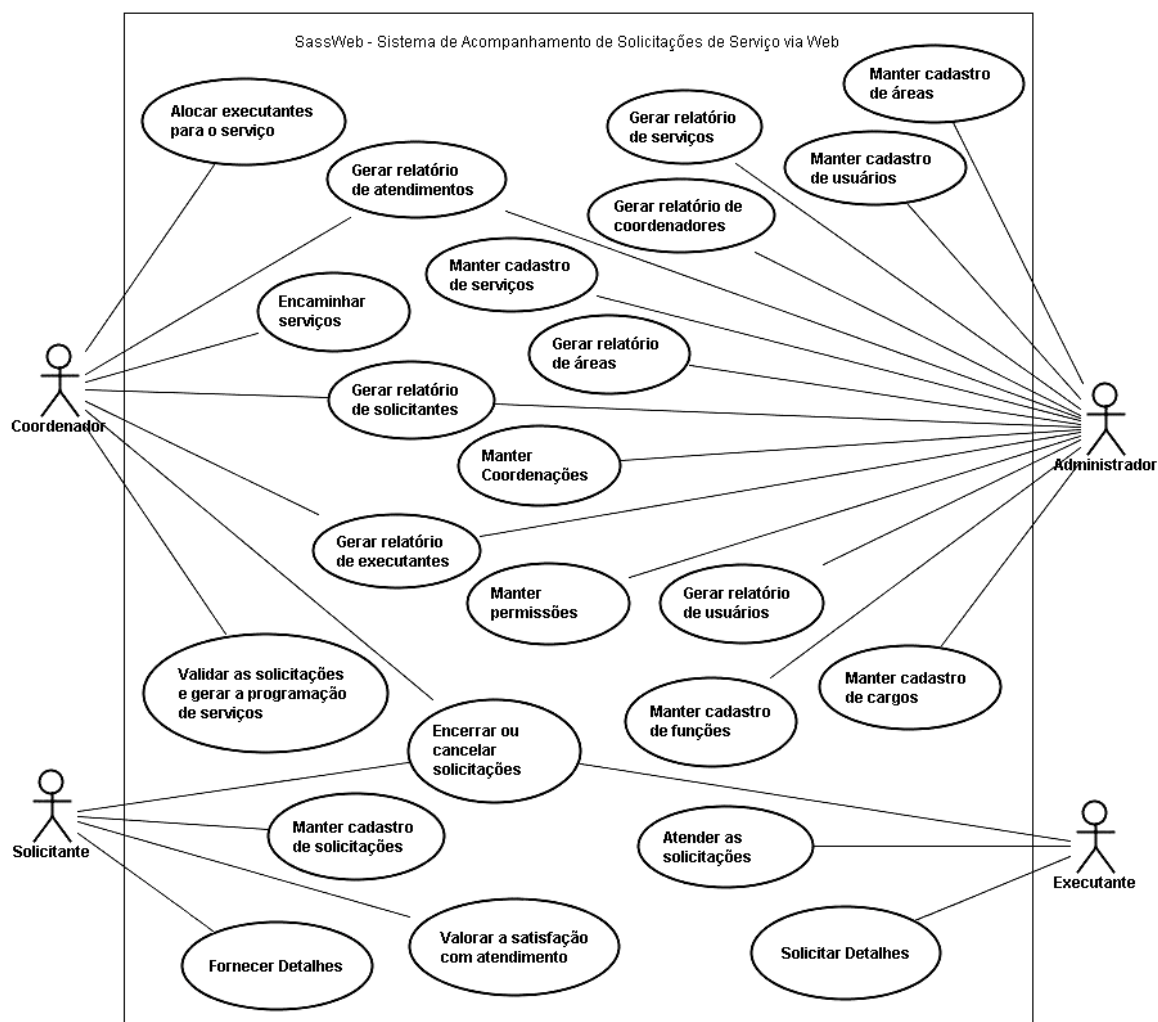
Cronograma de Atividades e Alocação de Recursos Humanos (prazo estimado em 45 dias)

1	2	3	4	9	20	25	27	30	40	45
A			D		F	H	J	K	N	P
B	C	E			G	I	L	M	O	Q

Atividades	Profissionais	Duração (dias)	HH x dias	A	P2	P1	Dependência
A Aquisição de requisitos (entrevista)	analista	3	2h x 3	6			—
B Instalação do ambiente	programador 1	1	4h x 1			4	—
C Aquisição de requisitos (entrevista)	programador 1	2	2h x 2			4	B
D Projeto do BD e arquitetura do sistema	analista	5	6h x 5	30			A,B
E Programação das telas	programador 1	5	6h x 5			30	C
F Especificações e documentação	analista	11	6h x 11	66			D
G Programação (protótipo)	programador 1	11	6h x 11			66	D,E
Programação (MVD e DAO)	programador 2	11	3h x 11		33		
H Documentação	analista	4	6h x 4	24			F,G
I Programação (versão preliminar)	programador 1	6	6h x 6			36	F,G
J Treinamento de usuários	analista	2	6h x 2	12			H
K Correção/ajuste na documentação	analista	3	6h x 3	18			I,J
L Correção/ajuste na versão preliminar	programador 1	3	6h x 3			18	J,K
M Programação (versão final)	programador 1	10	4h x 10			40	L
Programação (versão final MVC e DAO)	programador 2	10	2h x 10		20		
N Documentação e manual	analista	10	2h x 10	20			L
O Programação (versão definitiva)	programador 1	5	2h x 5			10	M,N
P Ajuste na documentação e manual	analista	5	2h x 5	10			M,N
Q Distribuição da versão definitiva	analista	1	4h x 1	4			P,O
	programador 1	1	4h x 1			4	
Total de Horas:				190	53	212	

## CAPÍTULO 2 – MODELO DE CASOS DE USO

### 2.1 - DIAGRAMA DE CASOS DE USO



### 2.2 - DESCRIÇÕES TEXTUAIS DOS ATORES

**Administrador** — Usuário responsável pela manutenção dos dados de funcionários (usuários e login), áreas de atuação, cargos, funções e cadastro de



serviços, com acesso às listas de serviços, coordenadores, solicitantes e executantes. O administrador não poderá alocar executantes, encaminhar serviços e encerrar ou cancelar solicitações, mas mantém as coordenações dos serviços.

**Coordenador** — Usuário responsável por alocar executantes, encaminhar serviços, validar as solicitações e programar a ordem de execução dos serviços. Só tem acesso aos dados da área que coordena, mas pode cancelar ou encerrar solicitações e consultar a programação de serviços.

**Executante** — Responsável pelo atendimento, executando os serviços. Pode solicitar mais informações quando necessário sobre detalhes e/ou condições do serviço ao solicitante. Pode cancelar ou encerrar as solicitações que lhe foram encaminhadas e, em caso de necessidade, executar o serviço do colega de equipe.

**Solicitante** — Usuário que solicita um serviço, com acesso à pesquisa das solicitações que gerou. Pode alterar as informações que detalham as condições do serviço solicitado e valorar o atendimento. Pode cancelar ou encerrar suas solicitações não encaminhadas e/ou pendentes de atendimento. Trata-se de um funcionário validado, logado no sistema com qualquer nível de permissão.

## 2.3 - DESCRIÇÕES TEXTUAIS DOS CASOS DE USO

### 2.3.1 - UC01 - Manter Cadastro de Funcionários

**Sumário:** O administrador acessa o cadastro de funcionários da empresa para inclusão, alteração ou consulta.

**Ator Primário:** Administrador

**Fluxo Principal:**

1. O administrador acessa o cadastro de funcionários.
2. O sistema apresenta as operações: inclusão, alteração e consulta.
3. O administrador executa a operação desejada ou abandona o cadastro.
4. Se é inclusão ou alteração, o sistema verifica o preenchimento e se não há duplicidade de usuário, então salva.

**Fluxo Alternativo (3):** Inclusão

- a. O administrador requisita a inclusão de novo funcionário.
- b. O sistema gera um novo código e aguarda a digitação dos dados.
- c. O administrador fornece os dados do funcionário.

**Fluxo Alternativo (3): Alteração**

- a. O administrador solicita consulta ao cadastro de funcionários.
- b. O sistema apresenta a lista de funcionários.
- c. O administrador seleciona um funcionário.
- d. O administrador altera os dados do funcionário, exceto seu código.

**Fluxo Alternativo (3): Consulta**

- a. O administrador solicita consulta ao cadastro de funcionários.
- b. O sistema apresenta a lista de funcionários.
- c. O administrador seleciona um funcionário.
- d. O sistema apresenta os dados do funcionário.

**Pós-Condição:** O funcionário foi inserido, ou seus dados foram alterados ou consultados.

**Regras de Negócio:** RN01, RN18.

2.3.2 - UC02 - Manter Permissões

**Sumário:** O administrador acessa o cadastro de áreas para inclusão, alteração ou consulta.

**Ator Primário:** Administrador

- Fluxo Principal:**
1. O administrador acessa o cadastro de permissões.
  2. O sistema apresenta as operações: inclusão, a alteração e consulta.
  3. O administrador executa a operação desejada ou abandona o cadastro.
  4. Se é inclusão ou alteração, o sistema verifica o preenchimento e se não há duplicidade de permissão, então salva.

**Fluxo Alternativo (3): Inclusão**

- a. O administrador requisita a inclusão de nova permissão.
- b. O sistema gera um novo código e aguarda a digitação dos dados.
- c. O administrador fornece os dados da permissão.

**Fluxo Alternativo (3): Alteração**

- a. O administrador solicita consulta ao cadastro de permissões.
- b. O sistema apresenta a lista de permissões.
- c. O administrador seleciona uma permissão.
- d. O administrador altera os dados da permissão, exceto seu código.

**Fluxo Alternativo (3): Consulta**

- a. O administrador solicita consulta ao cadastro de permissões.
- b. O sistema apresenta a lista de permissões.
- c. O administrador seleciona uma permissão.
- d. O sistema apresenta os dados da permissão.

**Pós-Condição:** Uma permissão foi inserida ou seus detalhes foram alterados ou consultados.

**Regras de Negócio:** RN02, RN18, RN20.

### 2.3.3 - UC03 - Manter Cadastro de Áreas

**Sumário:** O administrador acessa o cadastro de áreas para inclusão, alteração ou consulta.

**Ator Primário:** Administrador

- Fluxo Principal:**
1. O administrador acessa o cadastro de áreas.
  2. O sistema apresenta as operações: inclusão, a alteração e consulta.
  3. O administrador executa a operação desejada ou abandona o cadastro.
  4. Se é inclusão ou alteração, o sistema verifica o preenchimento e se não há duplicidade de área, então salva.

**Fluxo Alternativo (3): Inclusão**

- a. O administrador requisita a inclusão de nova área.
- b. O sistema gera um novo código e aguarda a digitação dos dados.
- c. O administrador fornece os dados da área.

**Fluxo Alternativo (3): Alteração**

- a. O administrador solicita consulta ao cadastro de área.
- b. O sistema apresenta a lista de áreas.
- c. O administrador seleciona uma área.
- d. O administrador altera os dados da área, exceto seu código.

**Fluxo Alternativo (3): Consulta**

- a. O administrador solicita consulta ao cadastro de área.
- b. O sistema apresenta a lista de áreas.
- c. O administrador seleciona uma área.
- d. O sistema apresenta os dados da área.

**Pós-Condição:** Uma área foi inserida ou seus detalhes foram alterados ou consultados.

**Regras de Negócio:** RN18.

#### 2.3.4 - UC04 - Manter Cadastro de Cargos

**Sumário:** O administrador acessa o cadastro dos cargos para inclusão, alteração ou consulta.

**Ator Primário:** Administrador

**Fluxo Principal:**

1. O administrador acessa o cadastro de cargos.
2. O sistema apresenta as operações: inclusão, a alteração e consulta.
3. O administrador executa a operação desejada ou abandona o cadastro.
4. Se é inclusão ou alteração, o sistema verifica o preenchimento e se não há duplicidade de cargo, então salva.

**Fluxo Alternativo (3):** Inclusão

- a. O administrador requisita a inclusão de novo cargo.
- b. O sistema gera um novo código e aguarda a digitação dos dados.
- c. O administrador fornece os dados do cargo.

**Fluxo Alternativo (3):** Alteração

- a. O administrador solicita consulta ao cadastro de cargo.
- b. O sistema apresenta a lista de cargos.
- c. O administrador seleciona um cargo.
- d. O administrador altera os dados do cargo, exceto seu código.

**Fluxo Alternativo (3):** Consulta

- a. O administrador solicita consulta ao cadastro de cargo.
- b. O sistema apresenta a lista de cargos.
- c. O administrador seleciona um cargo.
- d. O sistema apresenta os dados do cargo.

**Pós-Condição:** O cargo foi inserido ou seus detalhes foram alterados ou consultados.

**Regras de Negócio:** RN18.

### 2.3.5 - UC05 - Manter Cadastro de Funções

**Sumário:** O administrador acessa o cadastro de funções para inclusão, alteração ou consulta.

**Ator Primário:** Administrador

**Fluxo Principal:**

1. O administrador acessa o cadastro de funções.
2. O sistema apresenta as operações: inclusão, a alteração e consulta.
3. O administrador executa a operação desejada ou abandona o cadastro.
4. Se é inclusão ou alteração, o sistema verifica o preenchimento e se não há duplicidade de função, então salva.

**Fluxo Alternativo (3):** Inclusão

- a. O administrador requisita a inclusão de nova função.
- b. O sistema gera um novo código e aguarda a digitação dos dados.
- c. O administrador fornece os dados da função.

**Fluxo Alternativo (3):** Alteração

- a. O administrador solicita consulta ao cadastro de função.
- b. O sistema apresenta a lista de funções.
- c. O administrador seleciona uma função.
- d. O administrador altera os dados da função, exceto seu código.

**Fluxo Alternativo (3):** Consulta

- a. O administrador solicita consulta ao cadastro de função.
- b. O sistema apresenta a lista de funções.
- c. O administrador seleciona uma função.
- d. O sistema apresenta os dados da função.

**Pós-Condição:** A função foi inserida ou seus detalhes foram alterados ou consultados.

**Regras de Negócio:** RN18.

### 2.3.6 - UC06 - Manter Coordenações

**Sumário:** O administrador acessa o cadastro dos coordenadores para inclusão, alteração ou consulta.

**Ator Primário:** Administrador

**Fluxo Principal:**

1. O administrador acessa o cadastro de coordenadores.

2. O sistema apresenta as operações: inclusão, a alteração e consulta.
3. O administrador executa a operação desejada ou abandona o cadastro.
4. Se é inclusão ou alteração, o sistema verifica o preenchimento e se não há duplicidade de coordenador, então salva.

**Fluxo Alternativo (3): Inclusão**

- a. O administrador requisita a inclusão de novo coordenador.
- b. O sistema gera um novo código e aguarda a digitação dos dados.
- c. O administrador fornece o novo coordenador.

**Fluxo Alternativo (3): Alteração**

- a. O administrador solicita consulta ao cadastro de coordenador.
- b. O sistema apresenta a lista de coordenadores.
- c. O administrador seleciona um coordenador.
- d. O administrador altera a situação do coordenador (ativo ou não) ou sua área.

**Fluxo Alternativo (3): Consulta**

- a. O administrador solicita consulta ao cadastro de coordenador.
- b. O sistema apresenta a lista de coordenadores.
- c. O administrador seleciona um coordenador.
- d. O sistema apresenta os dados do coordenador.

**Pós-Condição:** O coordenador foi inserido ou seus dados de coordenação foram alterados ou consultados.

**Regras de Negócio:** RN18.

2.3.7 - UC07 - Manter Cadastro de Serviços

**Sumário:** O administrador acessa o cadastro (inclusão, alteração e consulta) dos serviços.

**Ator Primário:** Administrador

**Fluxo Principal:**

1. O administrador acessa o cadastro de serviços.
2. O sistema apresenta as operações: inclusão, a alteração e consulta.
3. O administrador executa a operação desejada ou abandona o cadastro.

4. Se é inclusão ou alteração, o sistema verifica o preenchimento e se não há duplicidade de serviço, então salva.

**Fluxo Alternativo (3): Inclusão**

- a. O administrador requisita a inclusão de novo serviços.
- b. O sistema gera um novo código e aguarda a digitação dos dados.
- c. O administrador fornece os dados do serviço.

**Fluxo Alternativo (3): Alteração**

- a. O administrador solicita consulta ao cadastro de serviços.
- b. O sistema apresenta a lista de serviços.
- c. O administrador seleciona um serviço.
- d. O administrador altera os dados do serviço, exceto seu código.

**Fluxo Alternativo (3): Consulta**

- a. O administrador solicita consulta ao cadastro de serviços.
- b. O sistema apresenta a lista de serviços.
- c. O administrador seleciona um serviço.
- d. O sistema apresenta os dados do serviços

**Pós-Condição:** O serviço foi inserido ou seus detalhes foram alterados ou consultados.

**Regras de Negócio:** RN18.

### 2.3.8 - UC08 - Alocar Executantes para o Serviço

**Sumário:** O coordenador deverá atribuir um Executante disponível para a execução do serviço da sua área no prazo padrão definido.

**Ator Primário:** Coordenador

**Fluxo Principal:**

1. O coordenador consulta os serviços da sua área.
2. O sistema lista os serviços da área.
3. O coordenador seleciona o serviço.
4. O coordenador consulta as equipes (executantes livres e alocados).
5. O coordenador atribui um executante disponível para o serviço.

**Pós-Condição:** O coordenador aloca o executante para o serviço da sua área.

**Regras de Negócio:** RN06, RN14.

### 2.3.9 - UC09 – Validar as Solicitações e Gerar Programação de Serviços

**Sumário:** O coordenador valida a solicitação, verifica dependência de outra solicitação, encaminha para o executante do serviços e inclui a solicitação na programação.

**Ator Primário:** Coordenador

**Fluxo Principal:**

1. O coordenador consulta o serviço e verifica que é da sua área.
2. O coordenador verifica se a execução depende de outro serviço, consultando a programação.
3. O coordenador encaminha a solicitação.
4. O coordenador inclui o serviço na programação conforme sua prioridade e/ou dependência.

**Fluxo Alternativo (1):** Solicitação invalidada

- a. O coordenador não valida a solicitação, pois o serviço é de outra área, devolvendo-a ao solicitante.

**Fluxo Alternativo (2):** Solicitação dependente

- a. O coordenador atribui dependência à solicitação e faz a inclusão na programação.

**Pós-Condição:** O coordenador inclui a solicitação de serviço encaminhada na programação.

**Regras de Negócio:** RN05, RN14.

### 2.3.10 - UC10 - Encaminhar Serviços

**Sumário:** O coordenador valida a solicitação, verifica dependência de outra solicitação e encaminha para o executante do serviço.

**Ator Primário:** Coordenador

**Fluxo Principal:**

1. O coordenador consulta o serviço e verifica se existe executante alocado.
2. Se não há, o coordenador aloca um executante para o serviço.
3. O coordenador encaminha o serviço.

**Pós-Condição:** O coordenador encaminha a solicitação de serviço para o executante.

**Regras de Negócio:** RN07, RN14.



### 2.3.11 - UC11 - Manter Cadastro de Solicitações

**Sumário:** O solicitante realiza operações de cadastro (inclusão, alteração e consulta) das solicitações de serviço.

**Ator Primário:** Solicitante

**Fluxo Principal:**

1. O solicitante inicia uma solicitação de serviço.
2. O solicitante seleciona a área e o serviço desejados.
3. O solicitante preenche as informações/detalhes sobre o serviço.
4. O solicitante envia (grava) a solicitação.

**Fluxo Alternativo:** Consulta

- a. O solicitante aplica filtro/critério para consultar suas solicitações de serviço.
- b. O sistema apresenta a lista resultado da consulta.
- c. O solicitante seleciona uma solicitação.
- d. O sistema apresenta os dados da solicitação selecionada.

**Fluxo Alternativo:** Alteração

- a. O solicitante seleciona uma solicitação de serviço.
- b. O solicitante inclui mais informações sobre o serviço.
- c. O solicitante salva a alteração.

**Pós-Condição:** A solicitação de serviço foi inserida, seus detalhes foram alterados ou sua situação evoluiu.

**Regras de Negócio:** RN04, RN12, RN15.

### 2.3.12 - UC12 – Atender as Solicitações

**Sumário:** O Executante realiza o atendimento e dá encerramento no serviço.

**Ator Primário:** Executante.

**Fluxo Principal:**

1. O executante consulta na programação as solicitações que lhe foram encaminhadas.
2. O executante seleciona a solicitação no início da fila.
3. O executante, após o atendimento, dá encerramento no serviço.
4. O novo estado da solicitação é salvo.

**Pós-Condição:** O serviço estará encerrado e a solicitação em estado de encerramento pelo solicitante.

**Regras de Negócio:** RN08, RN09, RN16.

### 2.3.13 - UC13 - Encerrar ou Cancelar as Solicitações

**Sumário:** O Coordenador, Solicitante ou Executante poderão encerrar ou cancelar as solicitações.

**Atores Primários:** Coordenador, Solicitante e Executante.

**Fluxo Principal:**

1. O ator consulta as solicitações.
2. O ator seleciona uma solicitação.
- 3a. Se é coordenador ou é executante, cancela ou encerra sem valoração a solicitação.
- 3b. Se é solicitante, cancela a solicitação pendente ou encerra a solicitação executada valorando o atendimento.
4. O novo estado da solicitação é salvo.

**Pós-Condição:** A solicitação estará encerrada sem valoração, encerrada com valoração ou cancelada.

**Regras de Negócio:** RN10, RN14, RN15, RN16.

### 2.3.14 - UC14 – Valorar a Satisfação com o Atendimento

**Sumário:** O solicitante atribui nota ao atendimento após execução do serviço solicitado.

**Ator Primário:** Solicitante.

**Fluxo Principal:**

1. O solicitante consulta suas solicitações.
2. O solicitante seleciona uma solicitação atendida.
3. O solicitante atribui nota de 1 a 5 ao atendimento.
4. O novo estado da solicitação é salvo.

**Pós-Condição:** O atendimento estará valorado pelo solicitante, segundo as notas: 5 (ótimo), 4 (muito bom), 3 (bom), 2 (normal) e 1 (insatisfatório).

**Regras de Negócio:** RN10, RN15.

### 2.3.15 - UC15 – Solicitar Detalhes

**Sumário:** O Executante verifica que há necessidade de mais informações sobre o serviço.

**Ator Primário:** Executante.

- Fluxo Principal:**
1. O executante consulta na programação as solicitações que lhe foram encaminhadas.
  2. O executante seleciona a solicitação no início da fila.
  3. O executante verifica que a informação é insuficiente.
  4. O executante pede ao solicitante mais informações.
  5. O novo estado da solicitação é salvo.

**Pós-Condição:** São solicitadas mais informações sobre o serviço.

**Regras de Negócio:** RN16.

#### 2.3.16 - UC16 – Fornecer Detalhes

**Sumário:** O Solicitante fornece mais informações sobre o serviço.

**Ator Primário:** Solicitante.

- Fluxo Principal:**
1. O solicitante consulta suas solicitações.
  2. O solicitante seleciona uma solicitação encaminhada e/ou pendente.
  3. O solicitante inclui mais informações sobre o serviço.
  4. O novo estado da solicitação é salvo.

**Pós-Condição:** São fornecidas mais informações sobre o serviço.

**Regras de Negócio:** RN12, RN15.

#### 2.3.17 - UC17 – Gerar Relatório de Usuários

**Sumário:** O administrador emite o relatório de usuários.

**Ator Primário:** Administrador

- Fluxo Principal:**
1. O administrador aplica filtro/critério para a emissão do relatório.
  2. O sistema apresenta o resultado obtido de acordo com os critérios selecionados.

**Pós-Condição:** O relatório de usuários é emitido para o administrador.

#### 2.3.18 - UC18 – Gerar Relatório de Coordenadores

**Sumário:** O administrador emite o relatório de coordenadores.

**Ator Primário:** Administrador

- Fluxo Principal:**
1. O administrador aplica filtro/critério para a emissão do relatório.

2. O sistema apresenta o resultado obtido de acordo com os critérios selecionados.

**Pós-Condição:** O relatório de coordenadores é emitido para o administrador.

#### 2.3.19 - UC19 – Gerar Relatório de Solicitações

**Sumário:** O administrador ou o coordenador emite o relatório de serviços.

**Atores Primários:** Administrador e Coordenador

**Fluxo Principal:**

1. O ator aplica filtro/critério para a emissão do relatório.
2. O sistema apresenta o resultado obtido de acordo com os critérios selecionados.

**Pós-Condição:** O relatório de solicitantes é emitido para o administrador ou para o coordenador.

#### 2.3.20 - UC20 – Gerar Relatório de Executantes

**Sumário:** O administrador ou o coordenador emite o relatório de executantes.

**Atores Primários:** Administrador e Coordenador

**Fluxo Principal:**

1. O ator aplica filtro/critério para a emissão do relatório.
2. O sistema apresenta o resultado obtido de acordo com os critérios selecionados.

**Pós-Condição:** O relatório de executantes é emitido para o administrador ou para o coordenador.

#### 2.3.21 - UC21 – Gerar Relatório de Áreas

**Sumário:** O administrador emite o relatório de áreas.

**Ator Primário:** Administrador

**Fluxo Principal:**

1. O administrador aplica filtro/critério para a emissão do relatório.
2. O sistema apresenta o resultado obtido de acordo com os critérios selecionados.

**Pós-Condição:** O relatório de áreas é emitido para o administrador.

### 2.3.22 - UC22 – Gerar Relatório de Serviços

**Sumário:** O administrador emite o relatório de serviços.

**Ator Primário:** Administrador

**Fluxo Principal:**

1. O administrador aplica filtro/critério para a emissão do relatório.
2. O sistema apresenta o resultado obtido de acordo com os critérios selecionados.

**Pós-Condição:** O relatório de serviços é emitido para o administrador.

### 2.3.23 - UC23 – Gerar Relatório de Atendimentos

**Sumário:** O administrador ou o coordenador emite o relatório de atendimentos.

**Atores Primários:** Administrador e Coordenador

**Fluxo Principal:**

1. O ator aplica filtro/critério para a emissão do relatório.
2. O sistema apresenta o resultado obtido de acordo com os critérios selecionados.

**Pós-Condição:** O relatório de atendimentos é emitido para o administrador ou para o coordenador.



## 3.2 – DICIONÁRIO DAS CLASSES

### 3.2.1 - Area

Propriedade	Tipo	Descrição
codare	int	Código da área
desare	String	Descrição da área
sigare	String	Sigla da área

### 3.2.2 - Cargo

Propriedade	Tipo	Descrição
codcar	int	Código do cargo
descar	String	Descrição do cargo

### 3.2.3 – Funcao

Propriedade	Tipo	Descrição
codfnc	int	Código da função
desfnc	String	Descrição da função

### 3.2.4 – Permissao

Propriedade	Tipo	Descrição
codper	int	Código da permissão
desper	String	Descrição da permissão
nivper	char	Nível da permissão

### 3.2.5 - Funcionario

Propriedade	Tipo	Descrição
codfun	int	Código do funcionário
matfun	int	Matrícula do funcionário
nomfun	String	Nome do funcionário
idtfun	String	Identidade do funcionário
idtdte	String	Data da emissão da identidade
idtorg	String	Órgão emissor da identidade
emlfun	String	Email do funcionário
sitfun	char	Situação do funcionário (ativo ou inativo)
area	Area	Área do funcionário
cargo	Cargo	Cargo do funcionário
funcao	Funcao	Função exercida pelo funcionário
logusu	String	Login do usuário
senusu	String	Senha do usuário
permissao	ArrayList<Permissao>	Níveis de permissão do usuário

### 3.2.6 – Servico

Propriedade	Tipo	Descrição
codser	int	Código do serviço
titser	String	Título do serviço
desser	String	Descrição geral do serviço
ppaser	int	Prazo padrão de execução do serviço
priser	char	Prioridade do serviço
area	Area	Área responsável pelo serviço
sitser	char	Situação do serviço (ativo ou inativo)
coordenacao	ArrayList<Coordenador>	Coordenadores do serviço
equipe	ArrayList<Executante>	Equipe de executantes do serviço

### 3.2.7 – Solicitacao

Propriedade	Tipo	Descrição
codsol	int	Código da solicitação
servico	Servico	Serviço solicitado
solicitante	Solicitante	Usuário solicitante
usuarios	Funcionario[]	Usuários coordenador, executante e encerrante
sitsol	char	Situação da solicitação
datas	Date[]	Datas dos eventos da solicitação
detalhe	ArrayList<Detalhe>	Detalhes textuais da solicitação
presol	Solicitacao	Serviço pré-requisito da execução
datadol	Date	Previsão do encerramento
avasol	char	Avaliação do atendimento

### 3.2.8 - Detalhe

Propriedade	Tipo	Descrição
codsol	int	Código da solicitação
orddet	int	Ordem do detalhe
datev4	Date	Data da solicitação do detalhe
tx1det	String	Texto da solicitação do detalhe
datev5	Date	Data da alteração do detalhe
tx2det	String	Texto incluído no detalhe

### 3.2.9 - Programacao

Propriedade	Tipo	Descrição
ordent	int	Ordem de execução da solicitação

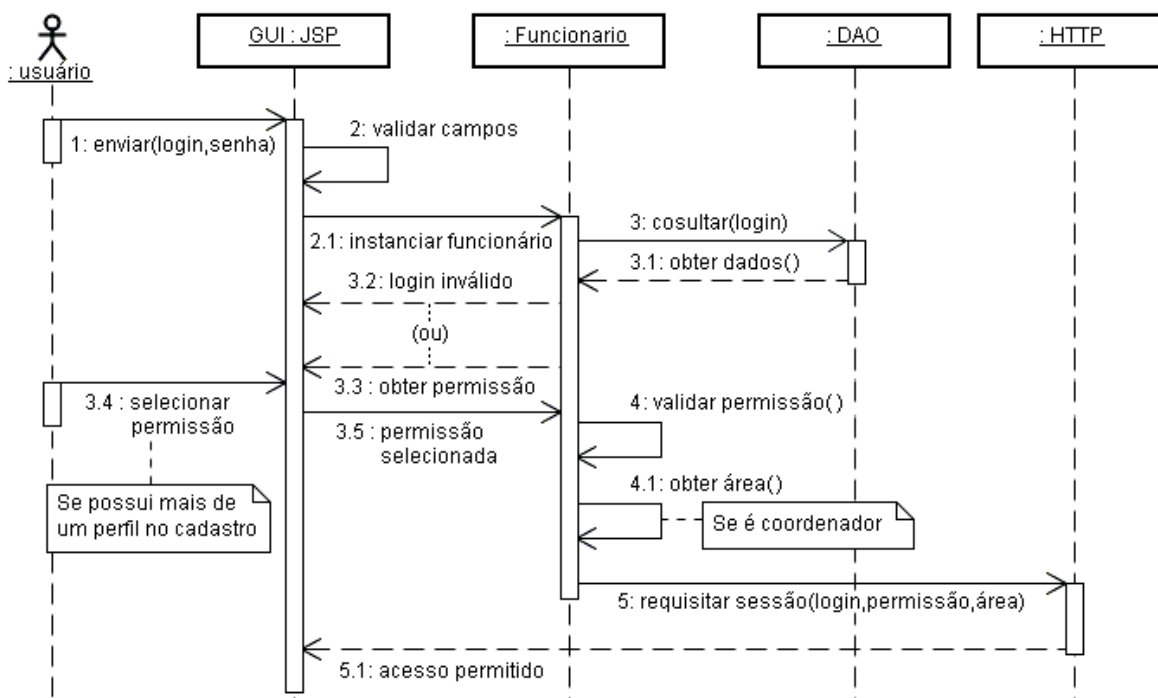


## CAPÍTULO 4 – MODELO DE INTERAÇÕES

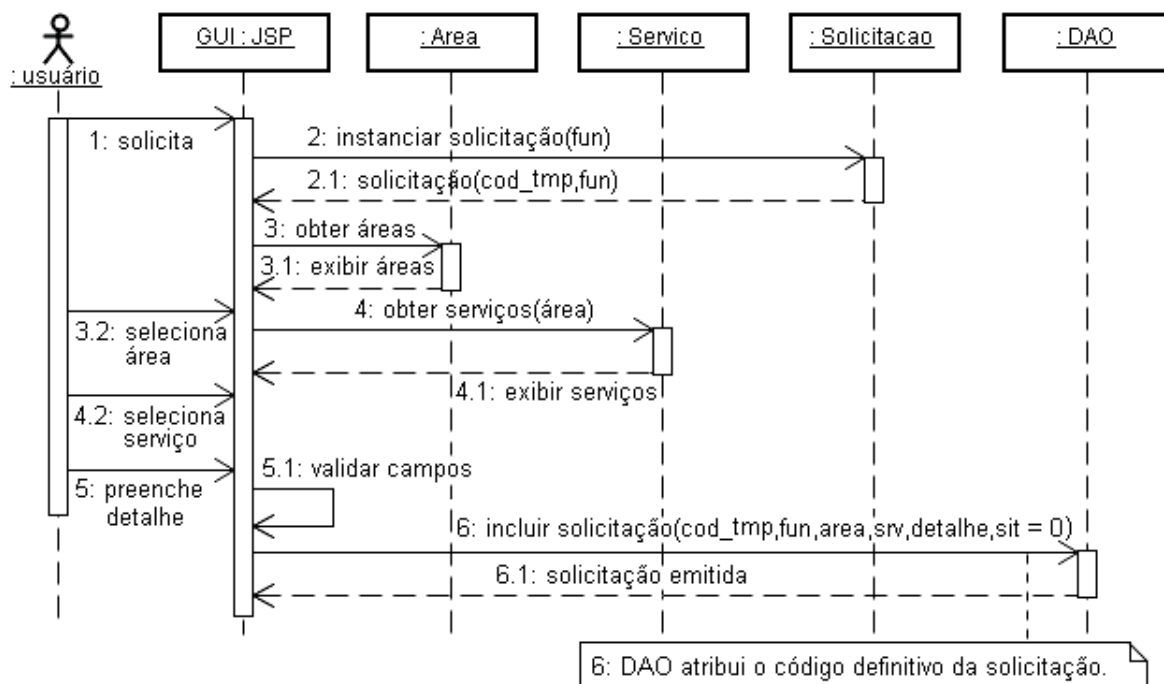
### 4.1 ---DIAGRAMAS DE SEQUÊNCIA

Optamos por representar a realização dos casos de uso através de diagramas de sequência, e não diagramas de colaboração, porque desejamos enfatizar a ordem temporal das mensagens. Nossos diagramas apresentam o cenário principal e o mínimo de detalhes necessários à sua representação e entendimento. Comentários foram adicionados onde necessário, para detalhar as alternativas, as condições e os critérios envolvidos no caso de uso realizado.

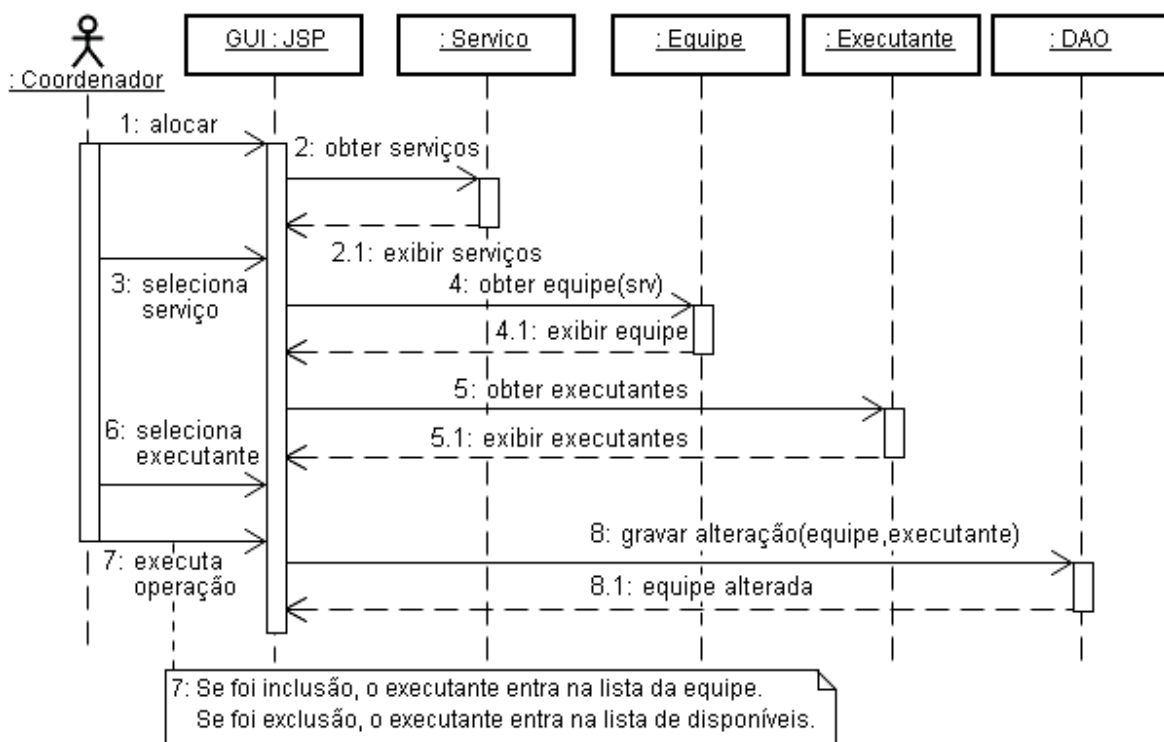
#### 4.1.1 – Efetuar Logon no Sistema



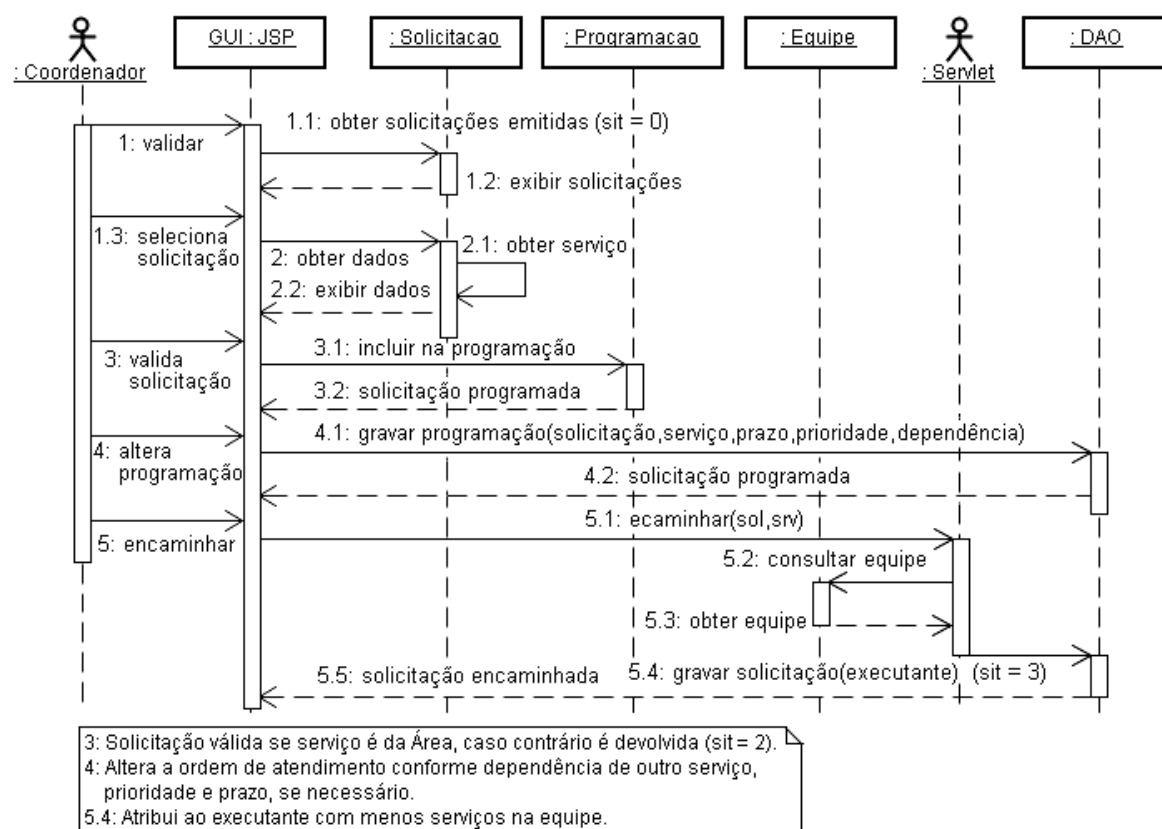
#### 4.1.2 – Emitir a Solicitação de Serviço



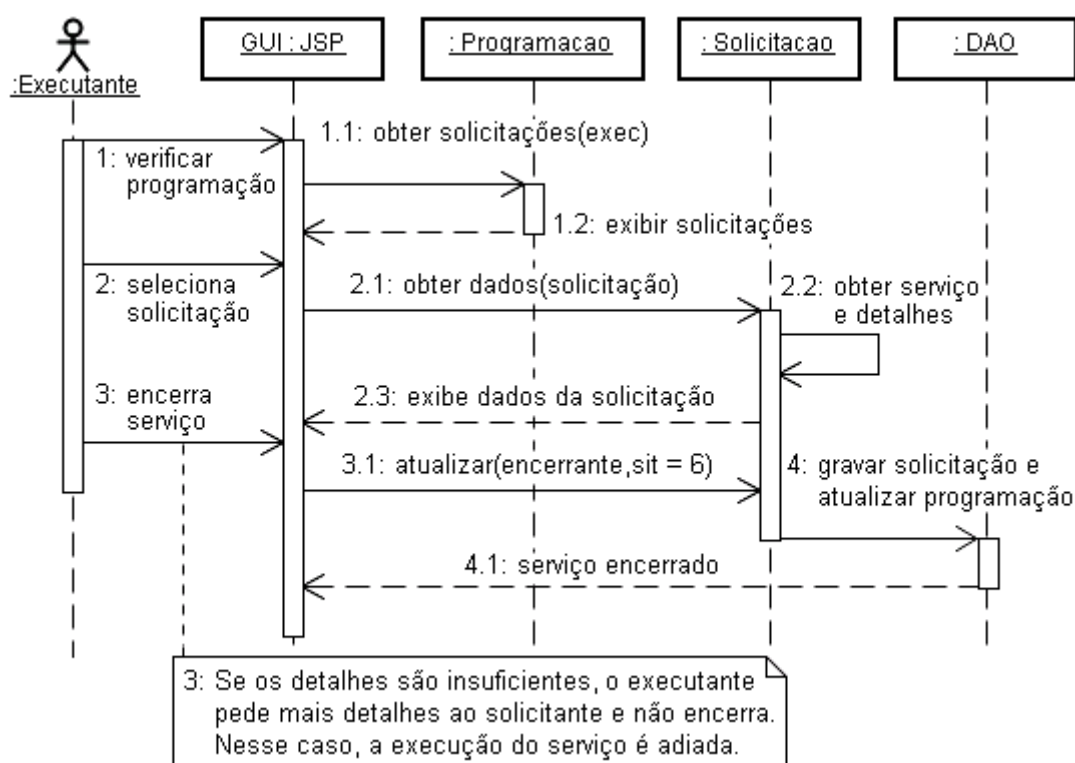
#### 4.1.3 – Atribuir o Executante ao Serviço



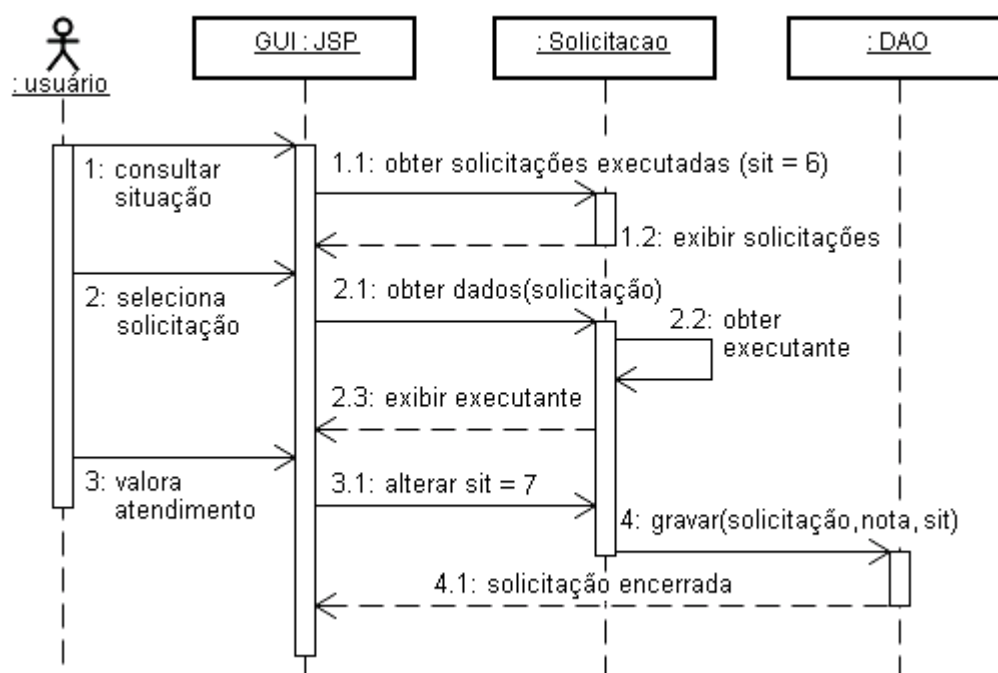
#### 4.1.4 – Encaminhar o Serviço



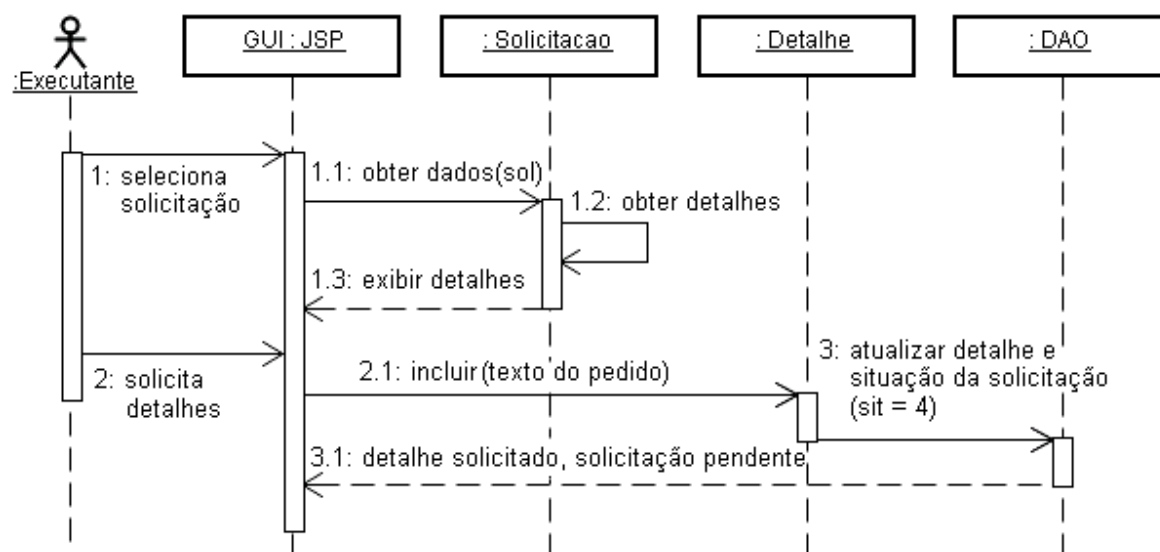
#### 4.1.5 – Encerrar o Serviço



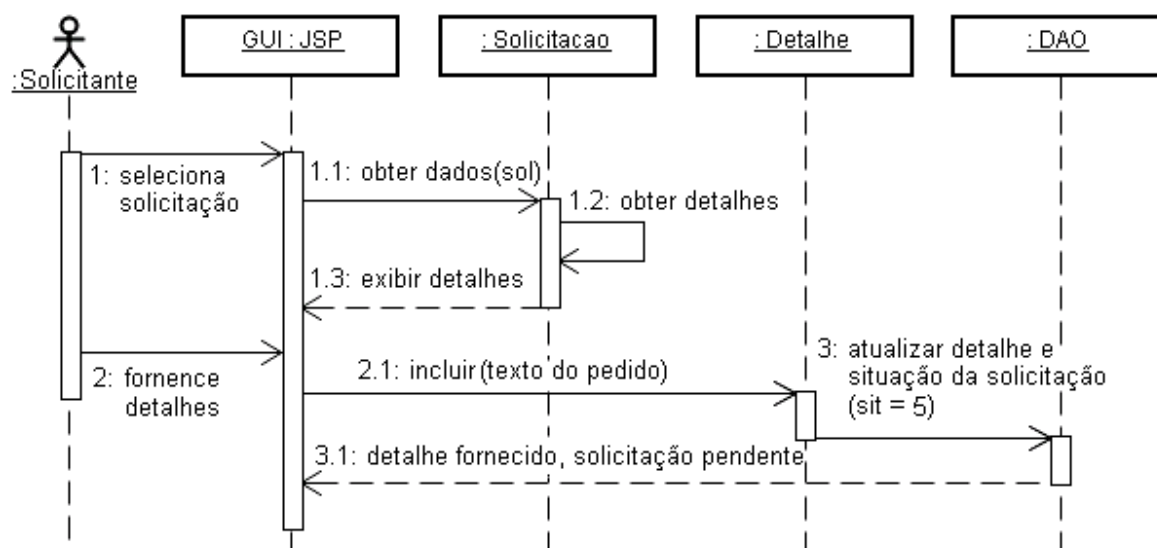
#### 4.1.6 – Encerrar a Solicitação



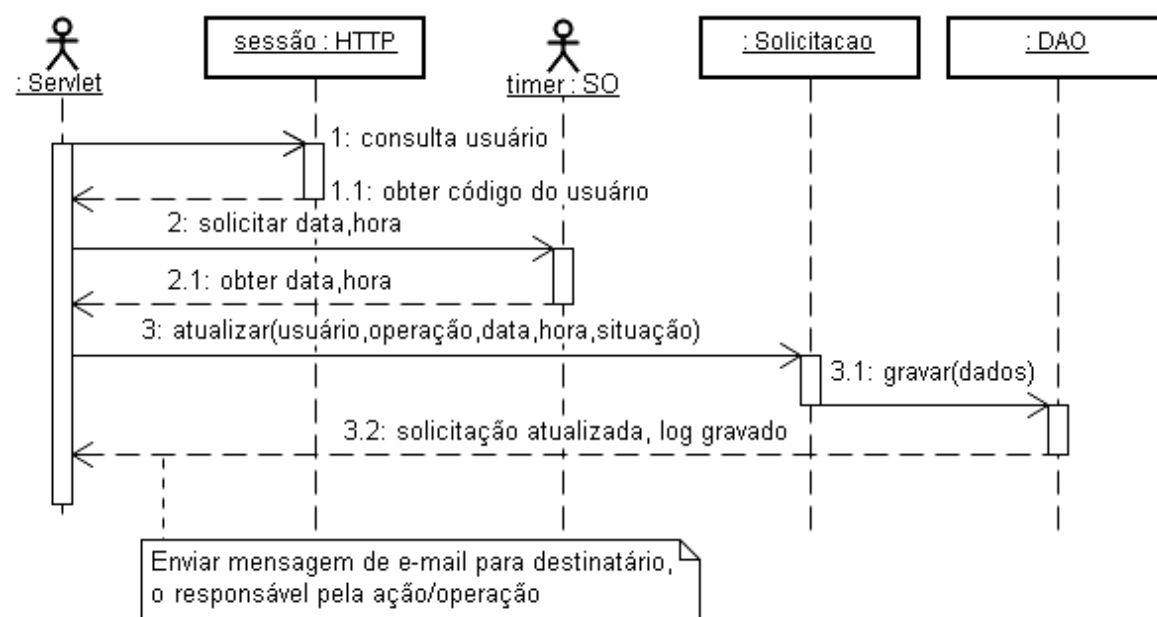
#### 4.1.7 – Solicitar Detalhes



#### 4.1.8 – Fornecer Detalhes



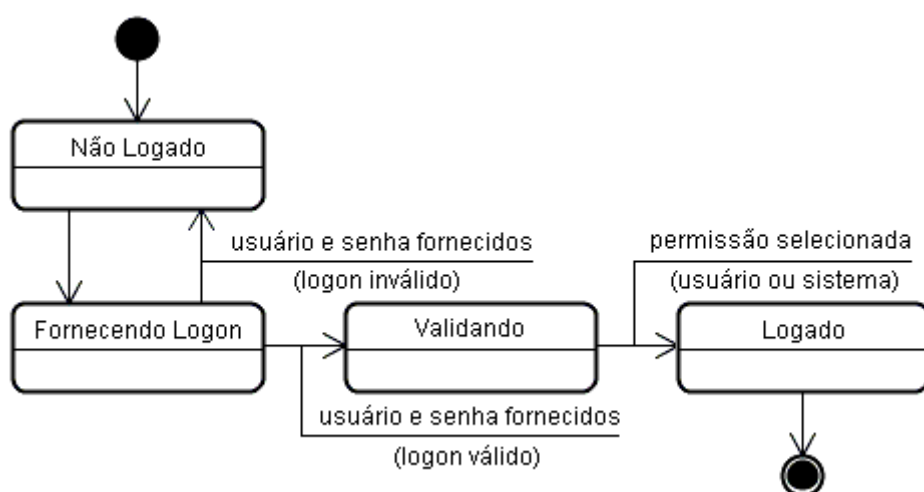
#### 4.1.9 – Gerar Log



## CAPÍTULO 5 – MODELO DE TRANSIÇÃO DE ESTADOS

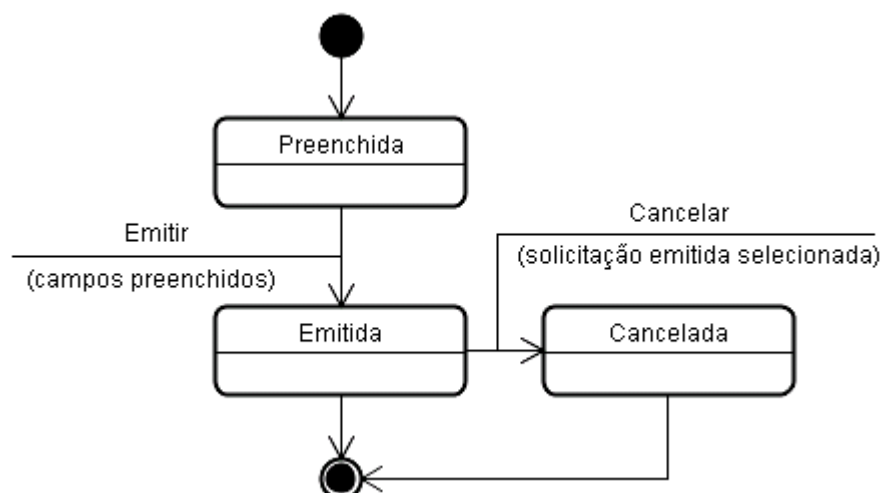
### 5.1 – FUNCIONÁRIO

#### 5.1.1– Efetuar Logon no Sistema

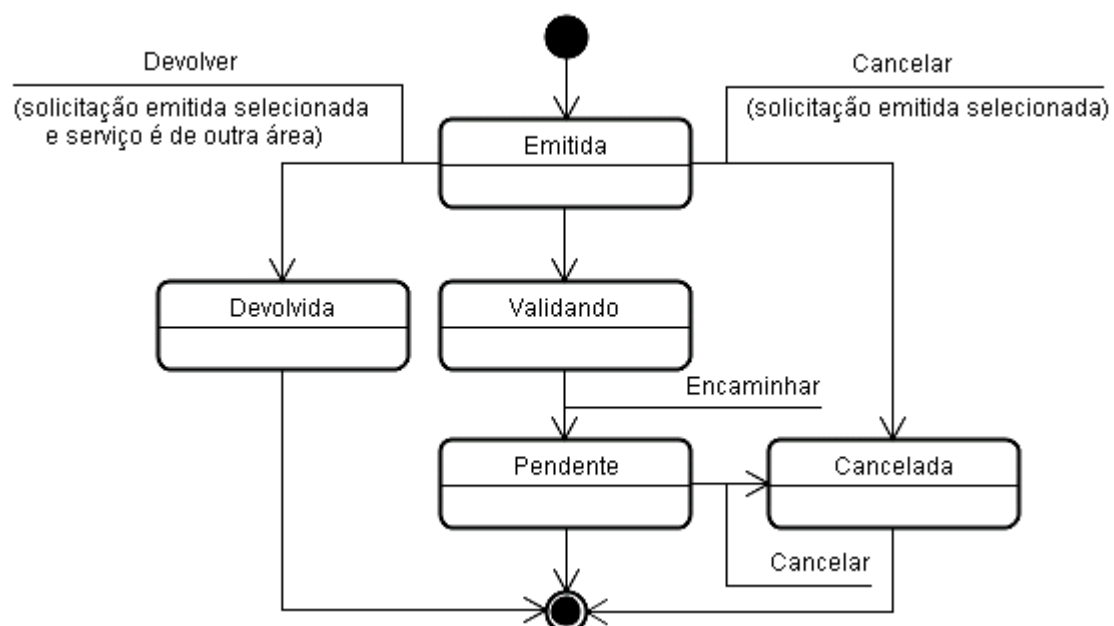


### 5.2 - SOLICITAÇÃO

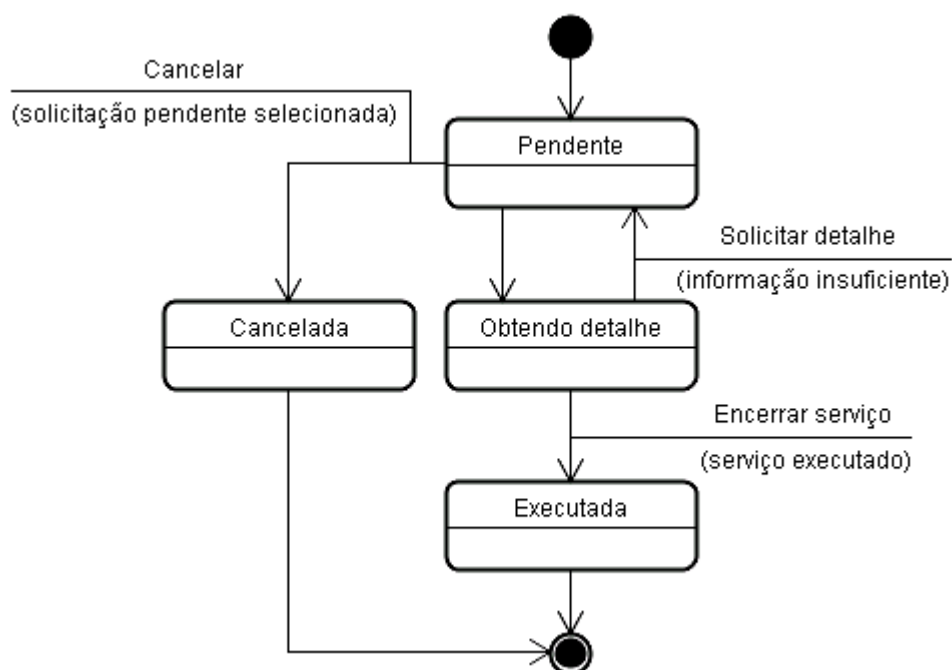
#### 5.2.1 – Emitir a Solicitação de Serviço



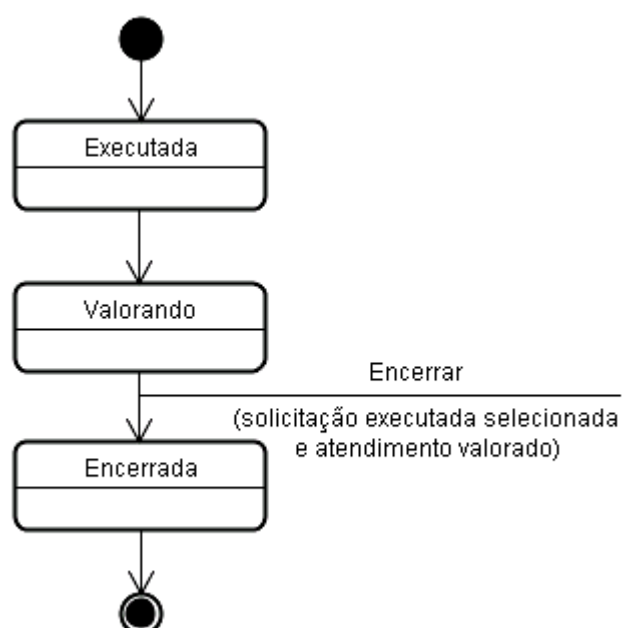
### 5.2.2 – Encaminhar o Serviço



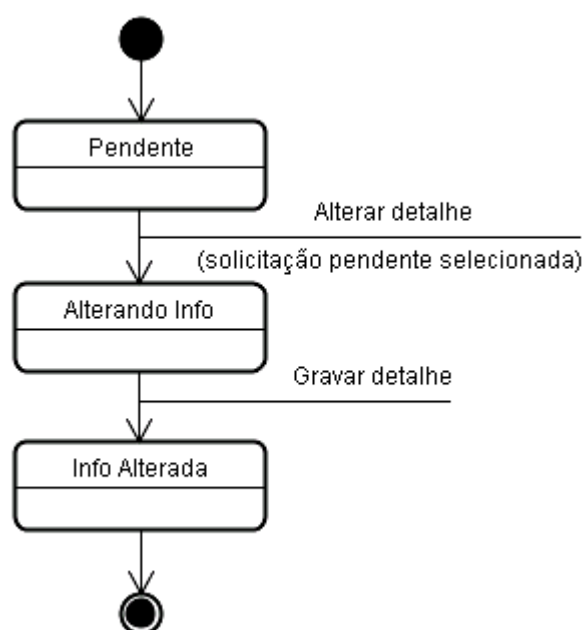
### 5.2.3 – Encerrar o Serviço



### 5.2.4 – Encerrar a Solicitação



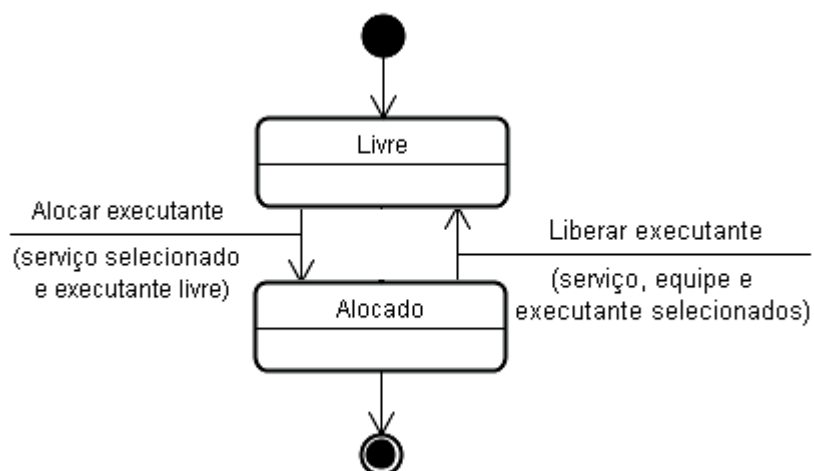
### 5.2.5 – Fornecer Detalhes





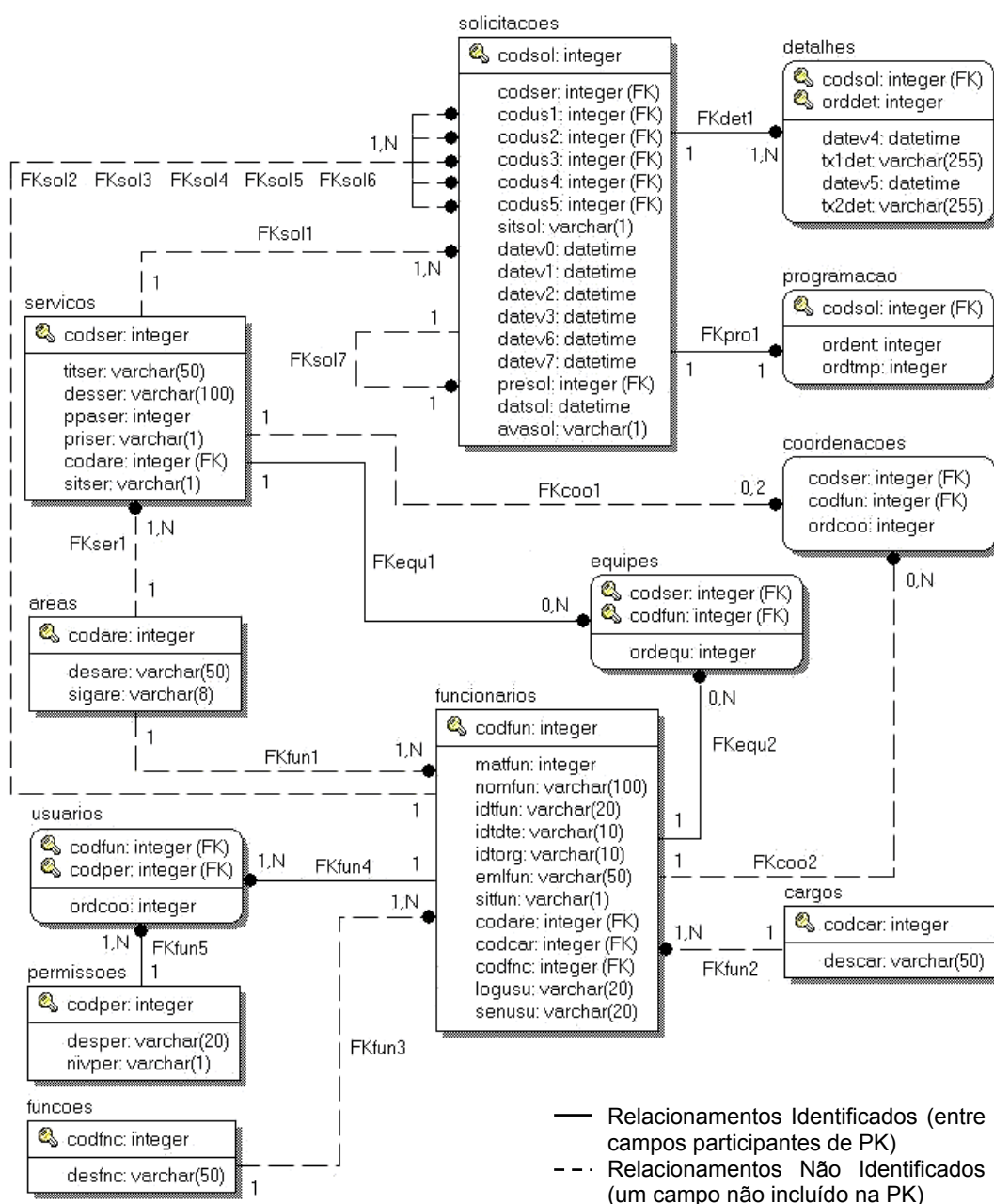
### 5.3 – EXECUTANTE

#### 5.3.1 – Atribuir Executante ao Serviço



## CAPÍTULO 6 – PROJETO DO BANCO DE DADOS

## 6.1 – PROJETO LÓGICO DO BANCO DE DADOS



## 6.2 – PROJETO FÍSICO DO BANCO DE DADOS

### Tipificação dos atributos, unicidade, obrigatoriedade e valor padrão.

#### 6.2.1. – Tabela Areas

Campo	Unicidade	Requerido	Tipo	Descrição	Padrão
codare	PK	sim	int(11)	Código da área	0
desare		sim	varchar(50)	Descrição da área	null
sigare		sim	varchar(8)	Sigla da área	null

#### 6.2.2 – Tabela Cargos

Campo	Unicidade	Requerido	Tipo	Descrição	Padrão
codcar	PK	sim	int(11)	Código do cargo	0
descar		sim	varchar(50)	Descrição do cargo	null

#### 6.2.3 – Tabela Funcoes

Campo	Unicidade	Requerido	Tipo	Descrição	Padrão
codfnc	PK	sim	int(11)	Código da função	0
desfnc		sim	varchar(50)	Descrição da função	null

#### 6.2.4 – Tabela Permissoes

Campo	Unicidade	Requerido	Tipo	Descrição	Padrão
codper	PK	sim	int(11)	Código da permissão	0
desper		sim	varchar(20)	Descrição da permissão	null
nivper		sim	char(1)	Nível hierárquico da permissão	0

#### 6.2.5 – Tabela Funcionarios

Campo	Unicidade	Requerido	Tipo	Descrição	Padrão
codfun	PK	sim	int(11)	Código do funcionário	0
matfun		sim	int(11)	Matrícula do funcionário	0
nomfun		sim	varchar(100)	Nome do funcionário	null
idtfun		sim	varchar(20)	Identidade do funcionário	null
idtdte		sim	varchar(10)	Data de emissão da identidade	null
idtorgr		sim	varchar(10)	Órgão emissor da identidade	null
emlfun		não	varchar(50)	Email do funcionário	null
sitfun		sim	char(1)	Situação do funcionário (Obs. 1)	1
codare	FK	sim	int(11)	Código da área do funcionário	0
coscar	FK	sim	int(11)	Código do cargo do funcionário	0
codfnc	FK	sim	int(11)	Código da função exercida	0
logusu		sim	varchar(20)	Login do usuário	null
senusu		sim	varchar(20)	Senha do usuário	null

Obs. 1: Valor 0 = funcionário inativo ou 1 = funcionário ativo.

#### 6.2.6 – Tabela Usuarios

Campo	Unicidade	Requerido	Tipo	Descrição	Padrão
codfun	PK	sim	int(11)	Código do funcionário	0
codper	PK	sim	int(11)	Código da permissão	0

### 6.2.7 – Tabela Servicos

Campo	Unicidade	Requerido	Tipo	Descrição	Padrão
codser	PK	sim	int(11)	Código do serviço	0
titser		sim	varchar(60)	Título do serviço	null
desser		sim	varchar(100)	Descrição geral do serviço	null
ppaser		sim	int(11)	Prazo padrão do serviço (Obs. 2)	24
priser		sim	char(1)	Prioridade do serviço (Obs. 3)	5
codare	FK	sim	int(11)	Código da área responsável	0
sitser		sim	char(1)	Situação do serviço (Obs. 4)	1

Obs.: 2. Prazo total em horas para execução do serviço. 3. Valor variando de 1 a 5, sendo que 1 corresponde à máxima prioridade. 4. Valor 0 = serviço inativo ou 1 = serviço ativo.

### 6.2.8 – Tabela Coordenacoes

Campo	Unicidade	Requerido	Tipo	Descrição	Padrão
codser		sim	int(11)	Código do serviço	0
codfun		sim	int(11)	Código do coordenador	0
ordcoo		sim	int(11)	Ordem do coordenador (Obs. 5)	0

Obs.: 5. Essa ordem determina a prevalência do coordenador (principal, suplente) na coordenação. Os três campos são indexados mas não formam chave primária unitária ou múltipla.

### 6.2.9 – Tabela Equipes

Campo	Unicidade	Requerido	Tipo	Descrição	Padrão
codser	PK	sim	int(11)	Código do serviço	0
codfun	PK	sim	int(11)	Código do executante	0
ordequ		sim	int(11)	Ordem do executante (Obs. 6)	0

Obs.: 6. Essa ordem determina apenas a sequência de entrada do executante na equipe.

### 6.2.10 – Tabela Solicitacoes

Campo	Unicidade	Requerido	Tipo	Descrição	Padrão
codsol	PK	sim	int(11)	Código da solicitação	0
codser	FK	sim	int(11)	Código do serviço solicitado	0
codus1	FK	sim	int(11)	Código do solicitante	0
codus2	FK	sim	int(11)	Código do coordenador encaminhante	0
codus3	FK	sim	int(11)	Código do executante escalado	0
codus4	FK	sim	int(11)	Código do executante efetivo	0
codus5	FK	sim	int(11)	Código do encerrante da solicitação	0
sitsol		sim	char(1)	Situação da solicitação (Obs. 7)	0
datev0		não	datetime	Data da emissão da solicitação	null
datev1		não	datetime	Data do cancelamento da solicitação	null
datev2		não	datetime	Data da devolução da solicitação	null
datev3		não	datetime	Data do encaminhamento da solicitação	null
datev6		não	datetime	Data do término do serviço solicitado	null
datev7		não	datetime	Data do encerramento da solicitação	null
presol	FK	sim	int(11)	Pré-requisito da solicitação (Obs. 8)	0
datso1		não	datetime	Data prevista para o encerramento	null
avasol		sim	char(1)	Avaliação do atendimento (Obs. 9)	0

Obs.: 7. Mantém e informa a evolução da solicitação, sendo 0 = incluída, 1 = cancelada, 2 = devolvida (ou invalidada), 3 = encaminhada, 4 = mais informação foi solicitada, 5 = informação foi alterada, 6 = executada e 7 = encerrada. Nos estados 3, 4 e 5 a solicitação estará pendente.

8. É o código de uma solicitação contendo o pedido de um serviço que precisa ser executado antes como pré-requisito. O valor padrão é o código da própria solicitação. (auto-relacionamento).

9. Nota dada pelo solicitante ao atendimento recebido de 1 a 5, sendo que 5 = ótimo.

### 6.2.11 – Tabela Detalhes

<b>Campo</b>	<b>Unicidade</b>	<b>Requerido</b>	<b>Tipo</b>	<b>Descrição</b>	<b>Padrão</b>
codsol	PK	sim	int(11)	Código do detalhe (Obs. 10)	0
orddet	PK	sim	int(11)	Ordem do detalhe	0
datev4		não	datetime	Data da solicitação do detalhe	null
tx1det		não	varchar(255)	Texto da solicitação do detalhe	null
datev5		não	datetime	Data da inclusão do detalhe	null
tx2det		não	varchar(255)	Texto do detalhe incluído	null

Obs.: 10. O código do detalhe é igual ao código da solicitação à qual ele se refere. Para cada nova entrada, o par [codsol,0] corresponde à informação inicial obrigatória do solicitante e sua data [datev5] é a mesma da emissão da solicitação. Nesse registro, os campos [datev4] e [tx1det] ficam nulos.

### 6.2.12 – Tabela Programacao

<b>Campo</b>	<b>Unicidade</b>	<b>Requerido</b>	<b>Tipo</b>	<b>Descrição</b>	<b>Default</b>
codsol	PK	sim	int(11)	Código da solicitação	0
ordent		sim	int(11)	Ordem de entrada das solicitações	0
ordtmp		não	int(11)	Ordem temporária (auxiliar)	0

Obs.: O coordenador, ao validar uma solicitação em função da data da emissão, sua prioridade e dependência, pode determinar a ordem para a execução (programação) e faz o encaminhamento. O coordenador também verifica se há dependência de outro serviço já programado.

## 6.3 – *SCRIPTS DAS TABELAS DO BANCO DE DADOS*

### 6.3.1 – Tabela Areas

```
CREATE TABLE areas (
  codare int(11) NOT NULL default 0,
  desare varchar(50) NOT NULL,
  sigare varchar(8) NOT NULL,
  PRIMARY KEY (codare)
);
```

### 6.3.2 – Tabela Cargos

```
CREATE TABLE cargos (
  codcar int(11) NOT NULL default 0,
  descar varchar(50) NOT NULL,
  PRIMARY KEY (codcar)
);
```

### 6.3.3 – Tabela Funcoes

```
CREATE TABLE funcoes (
  codfnc int(11) NOT NULL default 0,
  desfnc varchar(50) NOT NULL,
  PRIMARY KEY (codfnc)
);
```

### 6.3.4 – Tabela Permissoes

```
CREATE TABLE permissoes (
  codper int(11) NOT NULL default 0,
  desper varchar(20) NOT NULL,
  nivper varchar(1) NOT NULL default 0,
  PRIMARY KEY (codper)
);
```

### 6.3.5 – Tabela Funcionarios

```
CREATE TABLE funcionarios (
    codfun int(11) NOT NULL default 0,
    matfun int(11) NOT NULL default 0,
    nomfun varchar(100) NOT NULL,
    idtfun varchar(20) NOT NULL,
    idtdte varchar(10) NOT NULL,
    idtorg varchar(10) NOT NULL,
    emlfun varchar(50) NOT NULL,
    sitfun varchar(1) NOT NULL default '1',
    codare int(11) NOT NULL default 0,
    codcar int(11) NOT NULL default 0,
    codfnc int(11) NOT NULL default 0,
    logusu varchar(20) NOT NULL,
    senusu varchar(20) NOT NULL default '123456',
    PRIMARY KEY (codfun),
    KEY FKfun1 (codare),
    KEY FKfun2 (codcar),
    KEY FKfun3 (codfnc),
    CONSTRAINT FKfun1 FOREIGN KEY (codare) REFERENCES areas (codare)
ON DELETE RESTRICT ON UPDATE CASCADE,
    CONSTRAINT FKfun2 FOREIGN KEY (codcar) REFERENCES cargos
(codcar) ON DELETE RESTRICT ON UPDATE CASCADE,
    CONSTRAINT FKfun3 FOREIGN KEY (codfnc) REFERENCES funcoes
(codfnc) ON DELETE RESTRICT ON UPDATE CASCADE
);
```

### 6.3.6 – Tabela Usuarios

```
CREATE TABLE usuarios (
    codfun int(11) NOT NULL default 0,
    codper int(11) NOT NULL default 0,
    PRIMARY KEY (codfun,codper),
    CONSTRAINT Fkdet4 FOREIGN KEY (codfun) REFERENCES funcionarios
(codfun) ON DELETE RESTRICT ON UPDATE CASCADE,
    CONSTRAINT Fkdet5 FOREIGN KEY (codper) REFERENCES permissoes
(codper) ON DELETE RESTRICT ON UPDATE CASCADE
);
```

### 6.3.7 – Tabela Servicos

```
CREATE TABLE servicos (
    codser int(11) NOT NULL default 0,
    titser varchar(60) NOT NULL,
    desser varchar(100) NOT NULL,
    ppaser int(11) NOT NULL default 24,
    priser varchar(1) NOT NULL default '5',
    codare int(11) NOT NULL default 0,
    sitser varchar(1) NOT NULL default '1',
    PRIMARY KEY (codser),
    KEY FKser1 (codare),
    CONSTRAINT FKser1 FOREIGN KEY (codare) REFERENCES areas (codare)
ON DELETE RESTRICT ON UPDATE CASCADE
);
```

### 6.3.8 – Tabela Coordenacoes

```
CREATE TABLE coordenacoes (
    codser int(11) NOT NULL default 0,
    codfun int(11) NOT NULL default 0,
    ordcoo smallint(6) NOT NULL default 0,
    KEY FKcool (codser),
    KEY FKcoo2 (codfun),
    KEY IDXcoo (ordcoo),
    CONSTRAINT FKcool FOREIGN KEY (codser) REFERENCES servicios
(codser) ON DELETE RESTRICT ON UPDATE CASCADE,
    CONSTRAINT FKcoo2 FOREIGN KEY (codfun) REFERENCES funcionarios
(codfun) ON DELETE RESTRICT ON UPDATE CASCADE
);
```

### 6.3.9 – Tabela Equipes

```
CREATE TABLE equipes (
    codser int(11) NOT NULL default 0,
    codfun int(11) NOT NULL default 0,
    ordequ int(11) NOT NULL default 0,
    PRIMARY KEY (codser,codfun),
    KEY IDXequ (ordequ),
    KEY FKequ2 (codfun),
    CONSTRAINT FKequ2 FOREIGN KEY (codfun) REFERENCES funcionarios
(codfun) ON DELETE RESTRICT ON UPDATE CASCADE,
    CONSTRAINT FKequ1 FOREIGN KEY (codser) REFERENCES servicios
(codser) ON DELETE RESTRICT ON UPDATE CASCADE
);
```

### 6.3.10 – Tabela Solicitacoes

```
CREATE TABLE solicitacoes (
    codsol int(11) NOT NULL default 0,
    codser int(11) NOT NULL default 0,
    codus1 int(11) NOT NULL default 0,
    codus2 int(11) NOT NULL default 0,
    codus3 int(11) NOT NULL default 0,
    codus4 int(11) NOT NULL default 0,
    codus5 int(11) NOT NULL default 0,
    sitsol varchar(1) NOT NULL default 0,
    datev0 datetime default NULL,
    datev1 datetime default NULL,
    datev2 datetime default NULL,
    datev3 datetime default NULL,
    datev6 datetime default NULL,
    datev7 datetime default NULL,
    presol int(11) NOT NULL default 0,
    datsol datetime,
    avasol varchar(1) NOT NULL default 0,
    PRIMARY KEY (codsol),
    KEY FKsol1 (codser),
    KEY FKsol2 (codus1),
    KEY FKsol3 (codus2),
    KEY FKsol4 (codus3),
    KEY FKsol5 (codus4),
    KEY FKsol6 (codus5),
    KEY FKsol7 (presol),
    CONSTRAINT FKsol1 FOREIGN KEY (codser) REFERENCES servicios
(codser) ON DELETE RESTRICT ON UPDATE CASCADE,
```

```

        CONSTRAINT Fksol2 FOREIGN KEY (codus1) REFERENCES funcionarios
(codfun) ON DELETE RESTRICT ON UPDATE CASCADE,
        CONSTRAINT Fksol3 FOREIGN KEY (codus2) REFERENCES funcionarios
(codfun) ON DELETE RESTRICT ON UPDATE CASCADE,
        CONSTRAINT Fksol4 FOREIGN KEY (codus3) REFERENCES funcionarios
(codfun) ON DELETE RESTRICT ON UPDATE CASCADE,
        CONSTRAINT Fksol5 FOREIGN KEY (codus4) REFERENCES funcionarios
(codfun) ON DELETE RESTRICT ON UPDATE CASCADE,
        CONSTRAINT Fksol6 FOREIGN KEY (codus5) REFERENCES funcionarios
(codfun) ON DELETE RESTRICT ON UPDATE CASCADE,
        CONSTRAINT Fksol7 FOREIGN KEY (presol) REFERENCES solicitacoes
(codsol) ON DELETE RESTRICT ON UPDATE CASCADE
    );

```

### 6.3.11 – Tabela Detalhes

```

CREATE TABLE detalhes (
    codsol int(11) NOT NULL default 0,
    orddet int(11) NOT NULL default 0,
    datev4 datetime,
    tx1det varchar(255),
    datev5 datetime,
    tx2det varchar(255),
    PRIMARY KEY (codsol,orddet),
    CONSTRAINT FKdet1 FOREIGN KEY (codsol) REFERENCES solicitacoes
(codsol) ON DELETE RESTRICT ON UPDATE CASCADE
);

```

### 6.3.12 – Tabela Programacao

```

CREATE TABLE programacao (
    codsol int(11) NOT NULL default 0,
    ordent int(11) NOT NULL default 0,
    ordtmp int(11) default 0,
    PRIMARY KEY (codsol),
    CONSTRAINT FKpro1 FOREIGN KEY (codsol) REFERENCES solicitacoes
(codsol) ON DELETE RESTRICT ON UPDATE CASCADE
);

```

### 6.3.13 – Tabelas Auxiliares

```

CREATE TABLE servicostemp (
    codser int(11) NOT NULL default 0,
    titser varchar(60) NOT NULL,
    desser varchar(100) NOT NULL,
    ppaser smallint(6) NOT NULL default 24,
    priser varchar(1) NOT NULL default '5',
    codare int(11) NOT NULL default 0,
    PRIMARY KEY (codser)
);

```

```

CREATE TABLE executantestemp (
    codfun int(11) default 0,
    codusu int(11) default 0,
    qtatd smallint(6) default 0,
    KEY codfun (codusu),
    KEY codfun1 (codfun),
    KEY ordequl (qtatd)
);

```



## 6.4 – *SCRIPTS DAS VIEWS DO BANCO DE DADOS*

### 6.4.1 – Consulta ListaFuncionarios

```
CREATE OR REPLACE ALGORITHM=UNDEFINED DEFINER=root@localhost
SQL SECURITY DEFINER VIEW listafuncionarios
AS select funcionarios.codfun AS codfun,
funcionarios.matfun AS matfun,
funcionarios.nomfun AS nomfun,
funcionarios.idtfun AS idtfun,
funcionarios.idtdte AS idtdet,
funcionarios.idtorg AS idtorg,
funcionarios.emlfun AS emlfun,
funcionarios.sitfun AS sitfun,
funcionarios.codare AS codare,
areas.desare AS desare,
areas.sigare AS sigare,
funcionarios.codcar AS codcar,
cargos.descar AS descar,
funcionarios.codfnc AS codfnc,
funcoes.desfnc AS desfnc,
usuarios.codper AS codper,
permissoes.desper AS desper
from (funcoes join (permissoes
join (cargos join (areas
join (funcionarios join usuarios
on((funcionarios.codfun = usuarios.codfun)))
on((areas.codare = funcionarios.codare)))
on((cargos.codcar = funcionarios.codcar)))
on((permissoes.codper = usuarios.codper)))
on((funcoes.codfnc = funcionarios.codfnc)))
where (funcionarios.codfun > 0);
```

### 6.4.2 – Consulta ListaServicos

```
CREATE OR REPLACE ALGORITHM=UNDEFINED DEFINER=root@localhost
SQL SECURITY DEFINER VIEW listaservicos
AS select servicos.codser AS codser,
servicos.titser AS titser,
servicos.desser AS desser,
servicos.ppaser AS ppaser,
servicos.priser AS priser,
servicos.sitser AS sitser,
servicos.codare AS codare,
areas.desare AS desare,
areas.sigare AS sigare
from (areas join servicos
on((areas.codare = servicos.codare)))
where (servicos.codser > 0);
```

### 6.4.3 – Consulta ListaCoordenadores

```
CREATE OR REPLACE ALGORITHM=UNDEFINED DEFINER=root@localhost
SQL SECURITY DEFINER VIEW listacoordenadores
AS select distinct funcionarios.codfun AS codfun,
funcionarios.nomfun AS nomfun
from (funcionarios join usuarios
on((funcionarios.codfun = usuarios.codfun)))
where ((funcionarios.codfun > 0)
and (usuarios.codper = 3));
```

#### 6.4.4 – Consulta ListaCoordenacoes

```
CREATE OR REPLACE ALGORITHM=UNDEFINED DEFINER=root@localhost
SQL SECURITY DEFINER VIEW listacoordenacoes
AS select coordenacoes.codser AS codser,
servicos.titser AS titser,
coordenacoes.ordcoo AS ordcoo,
coordenacoes.codfun AS codfun,
funcionarios.nomfun AS nomfun
from (servicos join (funcionarios
join coordenacoes
on((funcionarios.codfun = coordenacoes.codfun)))
on((servicos.codser = coordenacoes.codser)))
order by coordenacoes.codser,coordenacoes.ordcoo;
```

#### 6.4.5 – Consulta ConsultaExecutantes

```
CREATE OR REPLACE ALGORITHM=UNDEFINED DEFINER=root@localhost
SQL SECURITY DEFINER VIEW consultaexecutantes
AS select funcionarios.codfun AS codfun,
funcionarios.matfun AS matfun,
funcionarios.nomfun AS nomfun,
equipes.codser AS codser,
servicos.titser AS titser
from (servicos join (funcionarios
join equipes
on((funcionarios.codfun = equipes.codfun)))
on((servicos.codser = equipes.codser)))
where (funcionarios.matfun > 0);
```

#### 6.4.6 – Consulta ListaEquipes

```
CREATE OR REPLACE ALGORITHM=UNDEFINED DEFINER=root@localhost
SQL SECURITY DEFINER VIEW listaequipes
AS select equipes.codser AS codser,
servicos.titser AS titser,
equipes.codfun AS codfun,
funcionarios.matfun AS matfun,
funcionarios.nomfun AS nomfun
from (servicos join (funcionarios
join equipes
on((funcionarios.codfun = equipes.codfun)))
on((servicos.codser = equipes.codser)));
```

#### 6.4.7 – Consulta ListaExecutantes

```
CREATE OR REPLACE ALGORITHM=UNDEFINED DEFINER=root@localhost
SQL SECURITY DEFINER VIEW listaexecutantes
AS select funcionarios.codfun AS codfun,
funcionarios.nomfun AS nomfun
from (funcionarios join usuarios
on((funcionarios.codfun = usuarios.codfun)))
where ((funcionarios.codfun > 0)
and (usuarios.codper = 2));
```

#### 6.4.8 – Consulta ListaUsuarios

```
CREATE OR REPLACE ALGORITHM=UNDEFINED DEFINER=root@localhost
SQL SECURITY DEFINER VIEW listausuarios
AS select usuarios.codfun AS codfun,
usuarios.codper AS codper,
permissoes.desper AS desper,
permissoes.nivper AS nivper
from (permissoes join usuarios
on((permissoes.codper = usuarios.codper)));
```

#### 6.4.9 – Consulta ListaAreas

```
CREATE OR REPLACE ALGORITHM=UNDEFINED DEFINER=root@localhost
SQL SECURITY DEFINER VIEW listaareas
AS select distinct listaservicos.codare AS codare,
listaservicos.desare AS desare,
listaservicos.sigare AS sigare from listaservicos;
```

#### 6.4.10 – Consulta ConsultaSolicitacoes

```
CREATE OR REPLACE ALGORITHM=UNDEFINED DEFINER=root@localhost
SQL SECURITY DEFINER VIEW consultasolicitacoes
AS select solicitacoes.codsol AS codsol,
solicitacoes.codus1 AS codus1,
solicitacoes.codus2 AS codus2,
solicitacoes.codus3 AS codus3,
solicitacoes.codus2 AS codus4,
solicitacoes.codus3 AS codus5,
solicitacoes.codser AS codser,
servicos.titser AS titser,
servicos.ppaser AS ppaser,
servicos.codare AS codare,
areas.sigare AS sigare,
solicitacoes.sitsol AS sitsol,
solicitacoes.datev0 AS datev0,
solicitacoes.presol AS presol,
solicitacoes.datsol AS datsol,
solicitacoes.datev1 AS datev1,
solicitacoes.datev2 AS datev2,
solicitacoes.datev3 AS datev3,
solicitacoes.datev6 AS datev6
from (areas join (servicos join solicitacoes
on((servicos.codser = solicitacoes.codser)))
on((areas.codare = servicos.codare)));
```

#### 6.4.11 – Consulta ConsultaDetalhes

```
CREATE OR REPLACE ALGORITHM=UNDEFINED DEFINER=root@localhost
SQL SECURITY DEFINER VIEW consultadetalhes
AS select solicitacoes.codus1 AS codus1,
detalhes.codsol AS codsol,
detalhes.orddet AS orddet,
detalhes.datev4 AS datev4,
detalhes.txldet AS txldet,
detalhes.datev5 AS datev5,
detalhes.tx2det AS tx2det
from (solicitacoes join detalhes
on((solicitacoes.codsol = detalhes.codsol)));
```

#### 6.4.12 – Consulta AreaCoordenador

```
CREATE OR REPLACE ALGORITHM=UNDEFINED DEFINER=root@localhost
SQL SECURITY DEFINER VIEW areacoordenador
AS select coordenacoes.codser AS codser,
coordenacoes.codfun AS codfun,
servicos.codare AS codare,
areas.sigare AS sigare
from (areas join (servicos
join coordenacoes
on((servicos.codser = coordenacoes.codser)))
on((areas.codare = servicos.codare)));
```

#### 6.4.13 – Consulta ConsultaProgramacao

```
CREATE OR REPLACE ALGORITHM=UNDEFINED DEFINER=root@localhost
SQL SECURITY DEFINER VIEW consultaprogramacao
AS select programacao.codsol AS codsol,
programacao.ordent AS ordent,
solicitacoes.codus1 AS codus1,
solicitacoes.codus2 AS codus2,
solicitacoes.codus3 AS codus3,
solicitacoes.codus2 AS codus4,
solicitacoes.codus3 AS codus5,
solicitacoes.sitsol AS sitsol,
solicitacoes.datev0 AS datev0,
solicitacoes.datev1 AS datev1,
solicitacoes.datev2 AS datev2,
solicitacoes.datev3 AS datev3,
solicitacoes.datev6 AS datev6,
solicitacoes.datev7 AS datev7,
solicitacoes.presol AS presol,
solicitacoes.datsol AS datsol,
solicitacoes.codser AS codser,
servicos.titser AS titser,
servicos.ppaser AS ppaser,
servicos.priser AS priser,
servicos.codare AS codare,
areas.sigare AS sigare
from (areas join ((servicos
join solicitacoes on((servicos.codser = solicitacoes.codser)))
join programacao on((solicitacoes.codsol = programacao.codsol)))
on((areas.codare = servicos.codare)));
```

#### 6.4.14 – Consulta ContaAtendimentos

```
CREATE OR REPLACE ALGORITHM=UNDEFINED DEFINER=root@localhost
SQL SECURITY DEFINER VIEW contaatendimentos
AS select solicitacoes.codser AS codser,
solicitacoes.codus2 AS codus2,
solicitacoes.sitsol AS sitsol
from solicitacoes where ((solicitacoes.sitsol > '2')
and (solicitacoes.sitsol < '6'));
```

#### 6.4.15 – Consulta OrdemAtendimento

```
CREATE OR REPLACE ALGORITHM=UNDEFINED DEFINER=root@localhost
SQL SECURITY DEFINER VIEW ordematendimento
AS select consultaprogramacao.datsol AS datsol,
consultaprogramacao.priser AS priser,
consultaprogramacao.codsol AS codsol
from consultaprogramacao
order by consultaprogramacao.datsol,consultaprogramacao.priser,
consultaprogramacao.codsol;
```

#### 6.4.16 – Consulta ServicoEquipe

```
CREATE OR REPLACE ALGORITHM=UNDEFINED DEFINER=root@localhost
SQL SECURITY DEFINER VIEW servicoequipe
AS select equipes.codser AS codser,
equipes.codfun AS codfun,
servicos.codare AS codare,
areas.sigare AS sigare
from (areas join (servicos
join equipes on((servicos.codser = equipes.codser)))
on((areas.codare = servicos.codare))) where (equipes.codfun > 0);
```

#### 6.4.17 – Consulta ListaSolicitados

```
CREATE OR REPLACE ALGORITHM=UNDEFINED DEFINER=root@localhost
SQL SECURITY DEFINER VIEW listasolicitados
AS select solicitacoes.codser AS codser,
servicos.titser AS titser,
areas.codare AS codare
from (areas join (servicos join solicitacoes
on((servicos.codser = solicitacoes.codser)))
on((areas.codare = servicos.codare))) order by servicos.titser;
```

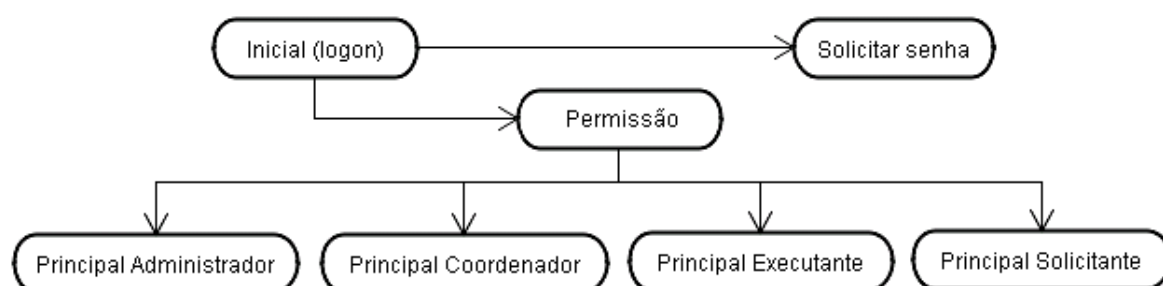
#### 6.4.18 – Consulta ListaAtendimentos

```
CREATE OR REPLACE ALGORITHM=UNDEFINED DEFINER=root@localhost
SQL SECURITY DEFINER VIEW listaatendimentos
AS select solicitacoes.codsol AS codsol,
solicitacoes.codser AS codser,
servicos.titser AS titser,
servicos.codare AS codare,
areas.sigare AS sigare,
solicitacoes.sitsol AS sitsol,
solicitacoes.codus1 AS codus1,
solicitacoes.codus1 AS codus2,
solicitacoes.codus1 AS codus3,
solicitacoes.codus1 AS codus4,
solicitacoes.codus1 AS codus5,
solicitacoes.datev0 AS datev0,
solicitacoes.datev1 AS datev1,
solicitacoes.datev2 AS datev2,
solicitacoes.datev3 AS datev3,
solicitacoes.datev6 AS datev6,
solicitacoes.datev7 AS datev7,
solicitacoes.presol AS presol,
solicitacoes.datsol AS datsol,
solicitacoes.avasol AS avasol
from (areas join (servicos join solicitacoes
on((servicos.codser = solicitacoes.codser)))
on((areas.codare = servicos.codare)));
```

## CAPÍTULO 7 – PROJETO DA INTERFACE GRÁFICA

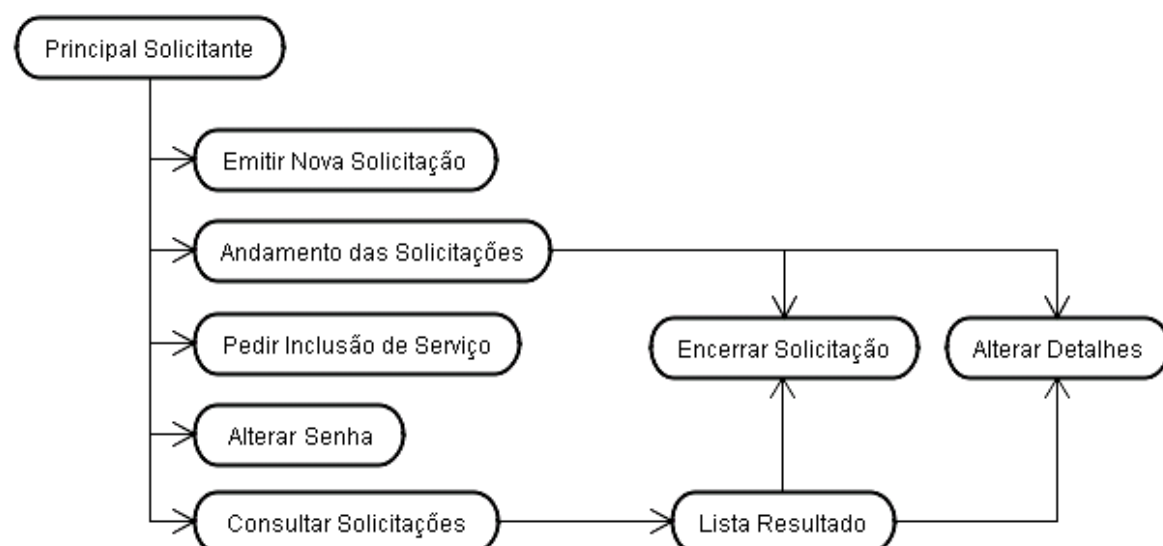
### 7.1 – HIERARQUIA DAS TELAS

#### 7.1.1 – Tela Inicial e Telas Principais



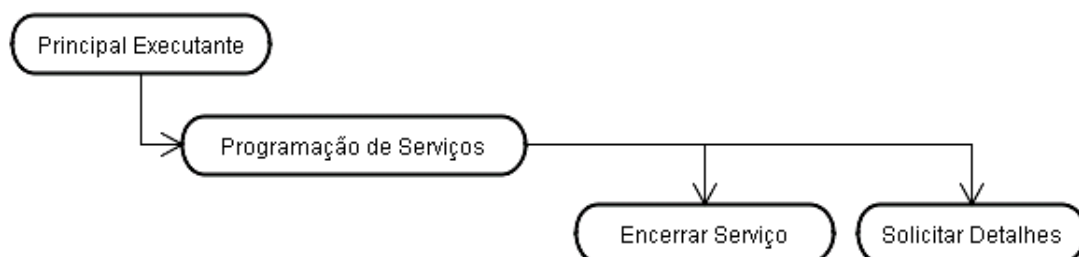
Obs.: Após o logon, o usuário entrará automaticamente na tela Principal correspondente à sua permissão.

#### 7.1.2 – Telas no Logon de Solicitante



Obs.: As telas do logon de Solicitante estão presentes em todos os outros logons e referem-se às solicitações que o usuário emitiu. Isso ocorre, porque todos podem solicitar serviços.

### 7.1.3 – Telas no Logon de Executante

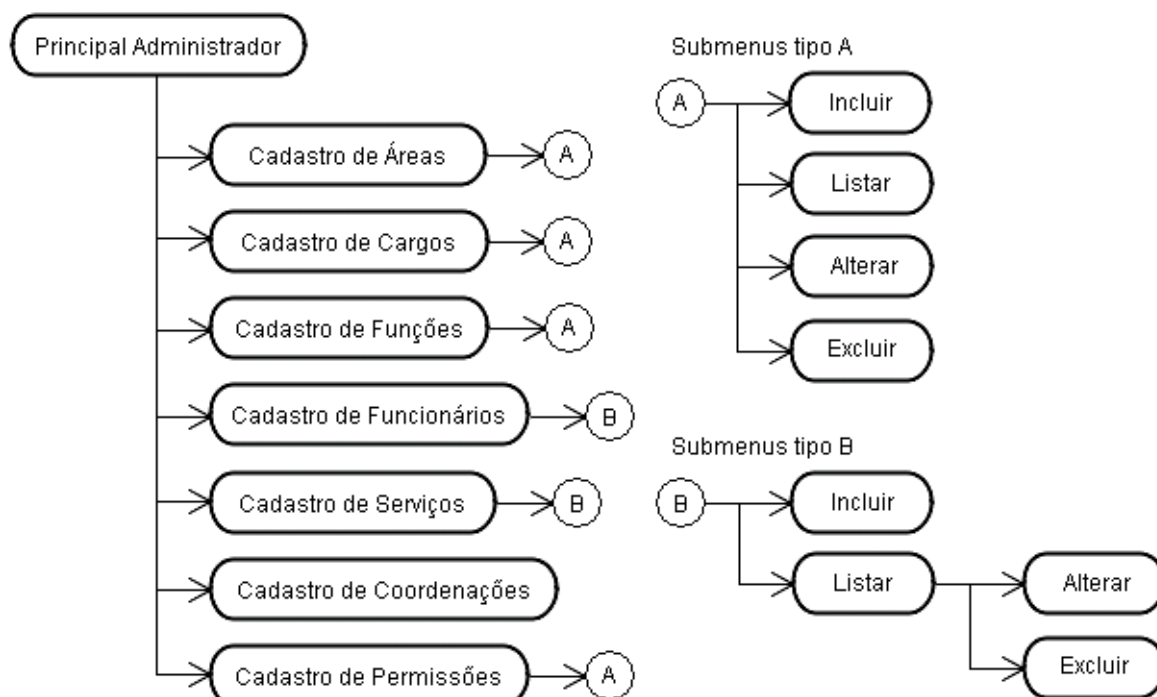


### 7.1.4 – Telas no Logon de Coordenador



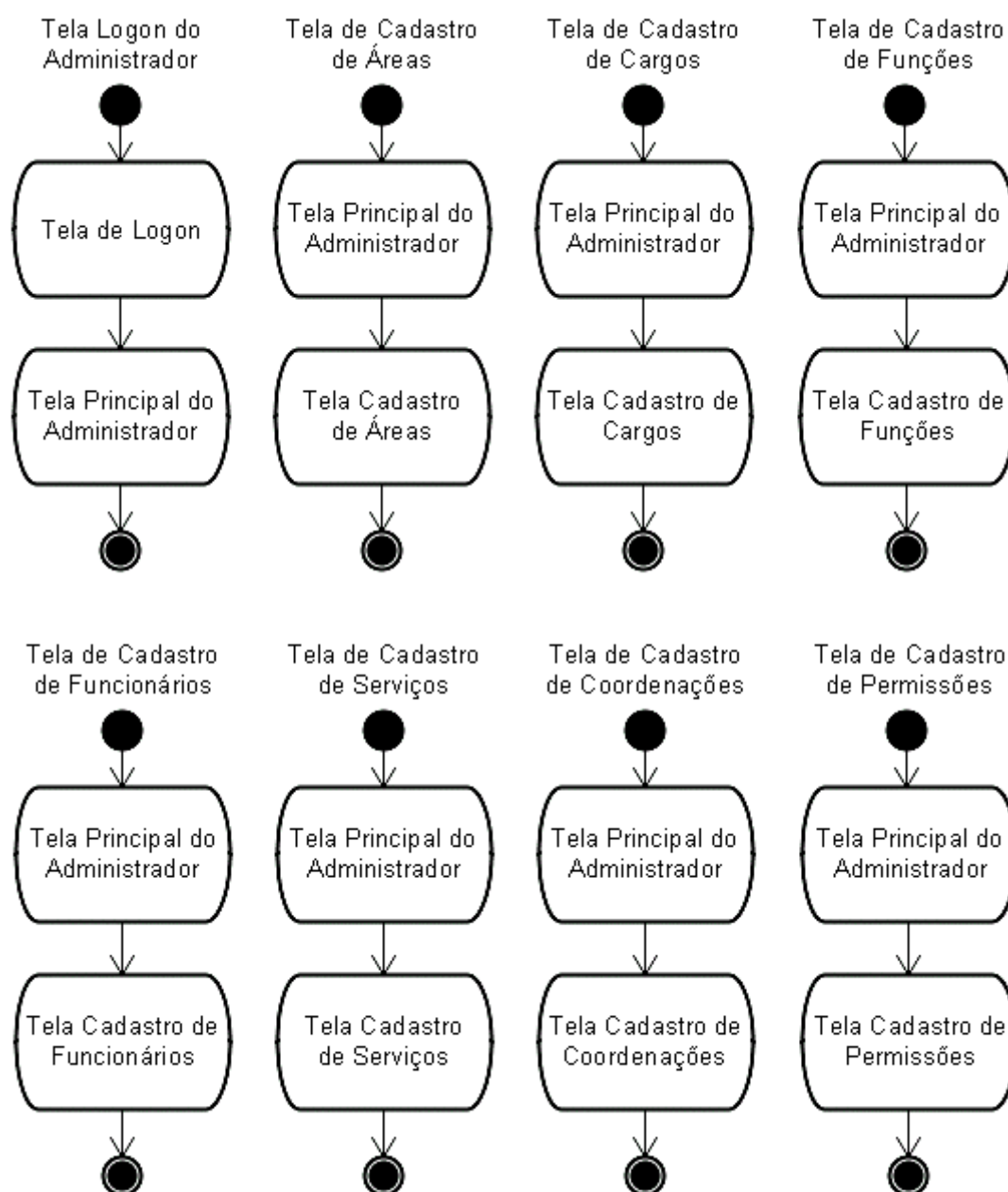
Obs.: As telas indicadas com (\*) estão presentes também no logon de Administrador.

### 7.1.5 – Telas no Logon de Administrador



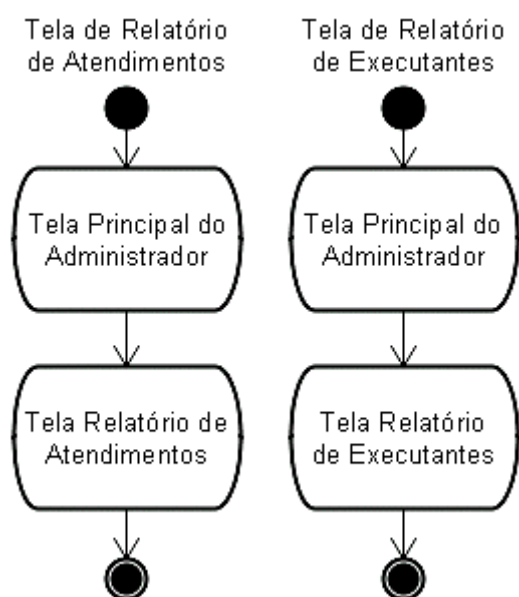
## 7.2 – MAPA DE NAVEGAÇÃO

### 7.2.1 – Logon de Administrador

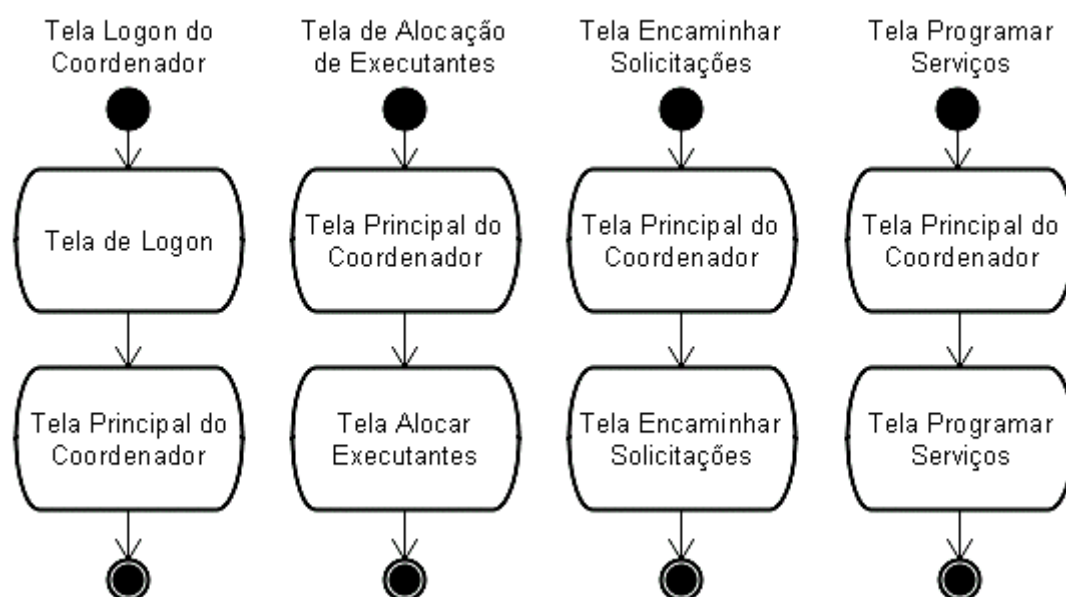


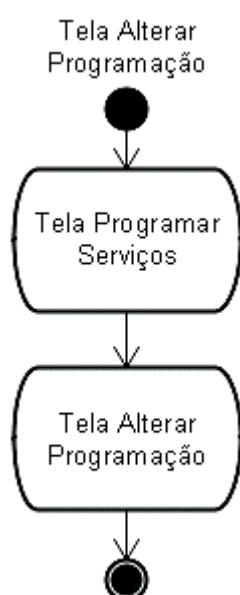


Obs.: Essas duas telas possuem correspondentes no logon de Coordenador.

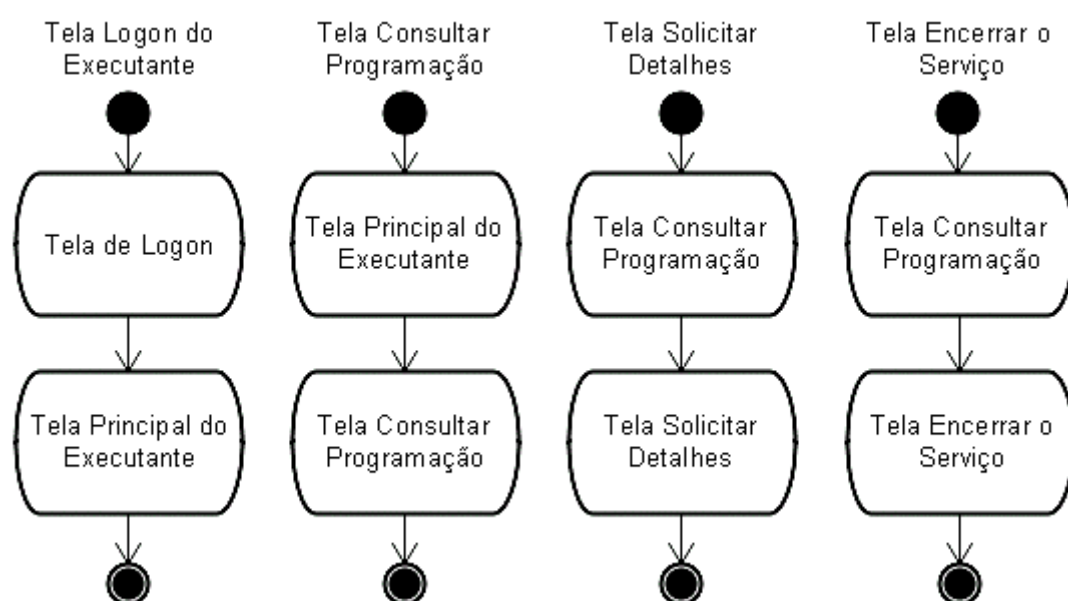


### 7.2.2 – Logon de Coordenador



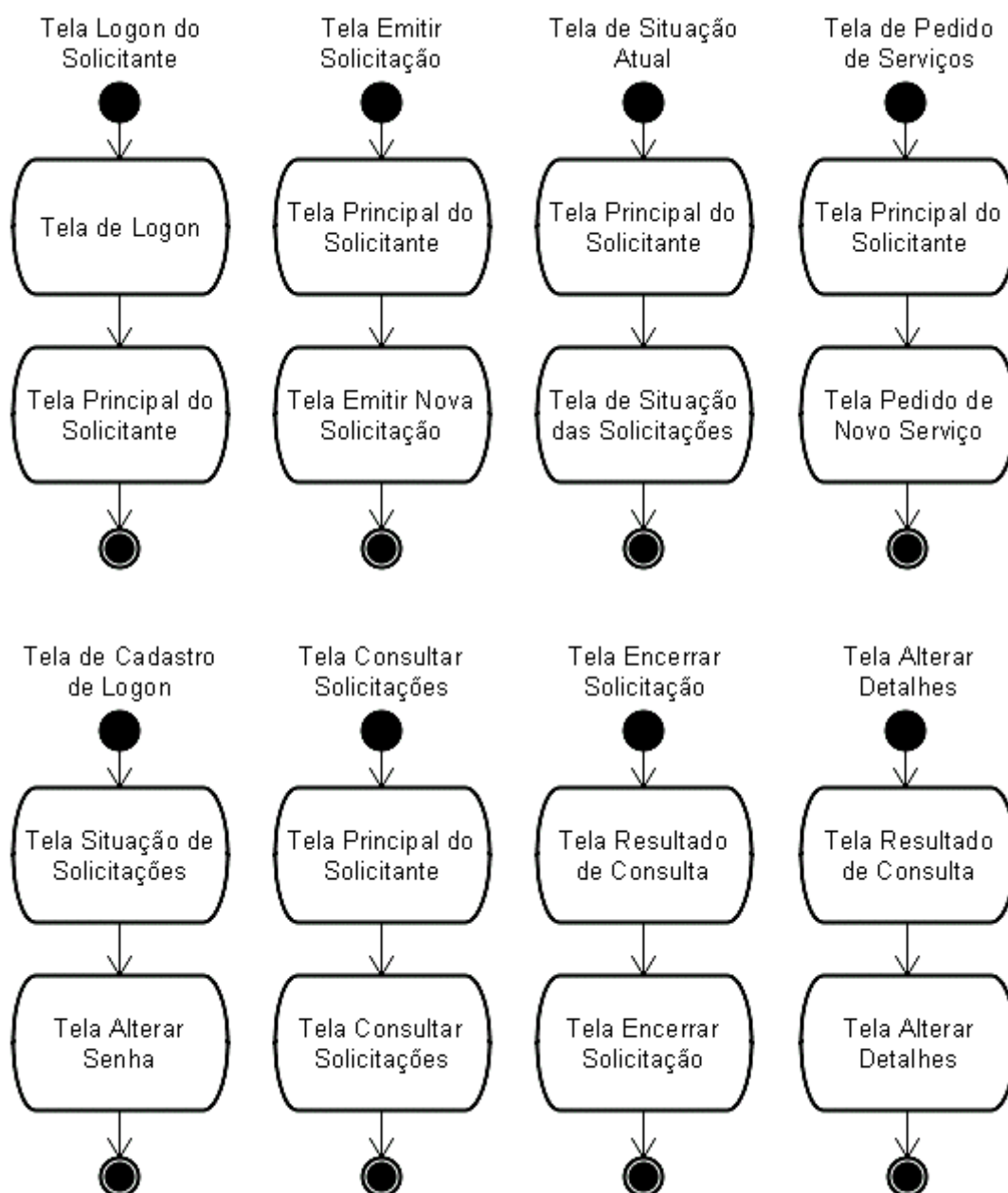


### 7.2.3 – Logon de Executante



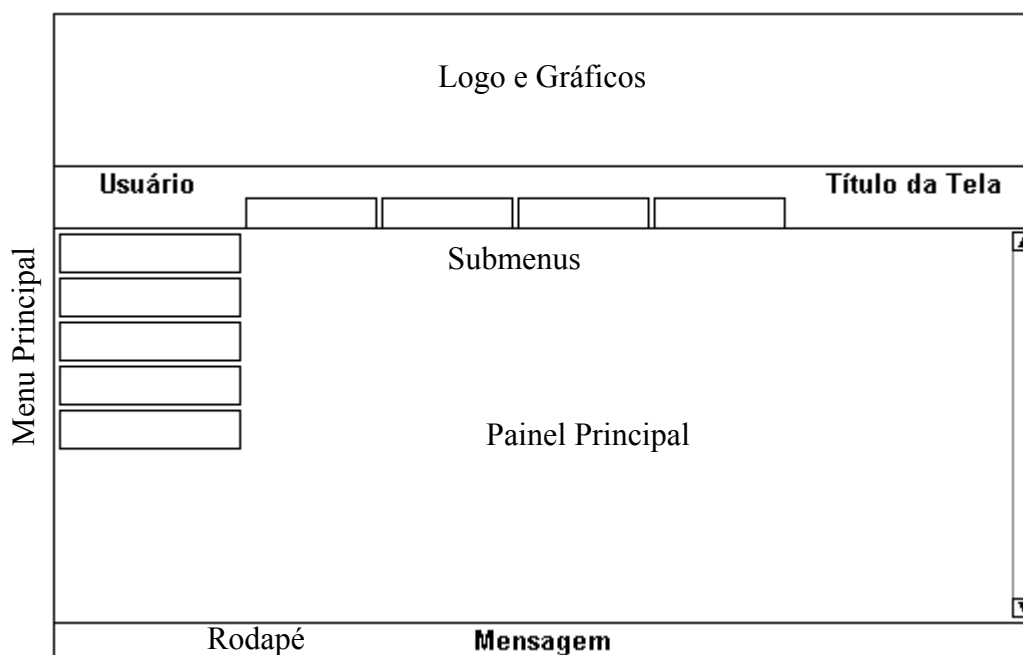
### 7.2.4 – Logon de Solicitante

Obs.: Essas telas estão presentes em todos os logons.



## 7.3 – PADRONIZAÇÃO DOS ELEMENTOS DA INTERFACE GRÁFICA

### 7.3.1 – Layout da Interface Gráfica



### 7.3.2 – Padronização dos Elementos da Interface

Descrição	Imagem
Opção do Menu Principal	
Opção do Submenu	
Link	
Botão	
Campo de Preenchimento	
Campo Não Editável	
Caixa de Listagem	
Campo de Texto do Sistema (não editável)	
Campo de Texto do Usuário (preenchimento)	
Botão de Seleção	
Ícone-botão Expandir	
Ícone-botão Editar / Alterar	
Ícone-botão Excluir	
Ícone-botão Imprimir	

## CAPÍTULO 8 – PROJETO DE IMPLEMENTAÇÃO

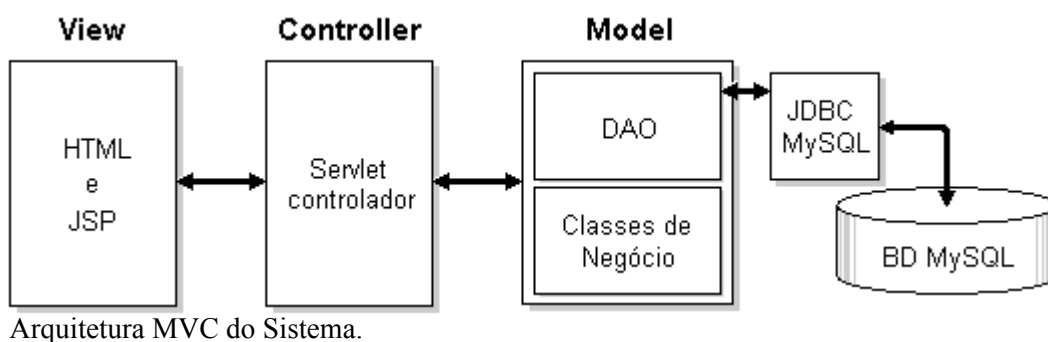
O primeiro passo será o projeto do Banco de Dados, sobre as definições das classes de modelo e regras de negócio, com a criação de tabelas e o estabelecimento dos relacionamentos entre elas, baseado nas informações coletadas nas entrevistas e nas especificações obtidas do DBA (Administrador do Banco de Dados da Empresa). Em seguida, faremos a definição das consultas e relatórios, mas não a sua implementação imediata. A próxima fase será a escolha da interface gráfica do usuário, sua formatação, cores, tamanhos, a disposição dos controles na tela, menus, botões, ícones e etc., que melhor se adaptar às características de utilização do sistema. Em seguida, os dados de funcionários, áreas, cargos e funções já existentes na empresa, serão carregados no Banco de Dados do nosso sistema.

Com a conclusão das etapas anteriores estará formada a base que irá apoiar o sistema propriamente dito, continuando com a definição das camadas MVC e a escrita das classes detalhadas no projeto. Nesse ponto, iniciaremos a programação em uma determinada ordem: logon, cadastros principais no logon de administrador, funcionalidades do logon de coordenador, funcionalidades dos logons de solicitante e executante. Finalmente, implementaremos os painéis de resultado de consultas, listagens e relatórios.

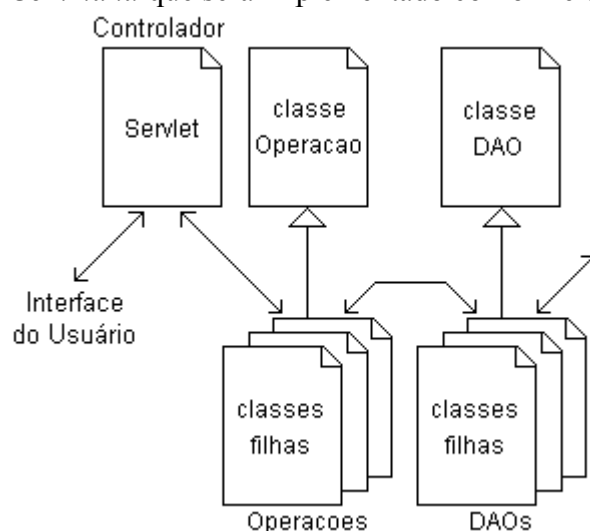
Na interface gráfica do usuário — GUI, utilizaremos JSP e Html com *frameset* para a construção das telas, possibilitando independência entre as diversas áreas de tela e o melhor desempenho nas atualizações dos formulários de cadastro e painéis de resultados. Sempre que possível o tratamento das entradas do usuário será feito na máquina cliente, através de JavaScript, limitando o fluxo cliente-servidor ao controle dos processos e acessos ao BD via Objeto de Acesso a Dados — DAO.

Na implementação do sincronismo dos processos paralelos do sistema — atualizar dados das solicitações, gravar log e enviar e-mail —, adotaremos a estratégia de atribuir multifuncionalidade à tabela *Solicitacoes*, ou seja, no momento em que uma atualização é feita, o log é gerado na mesma tabela e o sistema envia o e-mail de notificação logo em seguida. O uso desse recurso reduzirá o código e permitirá uma implementação simples e confiável.

A arquitetura de projeto utilizada será MVC, como especificado anteriormente, com implementação de padrão *Command*, encapsulando as funcionalidades de acesso a dados, realizadas via DAO e JDBC, segundo o diagrama esquemático das camadas apresentado abaixo.



As Classes de Negócio e Modelo serão aquelas definidas no Modelo de Classes, alocadas na pasta *pacoteSassweb*. O *Controller* utilizará o padrão *Command* que será implementado conforme o diagrama e observações seguintes.

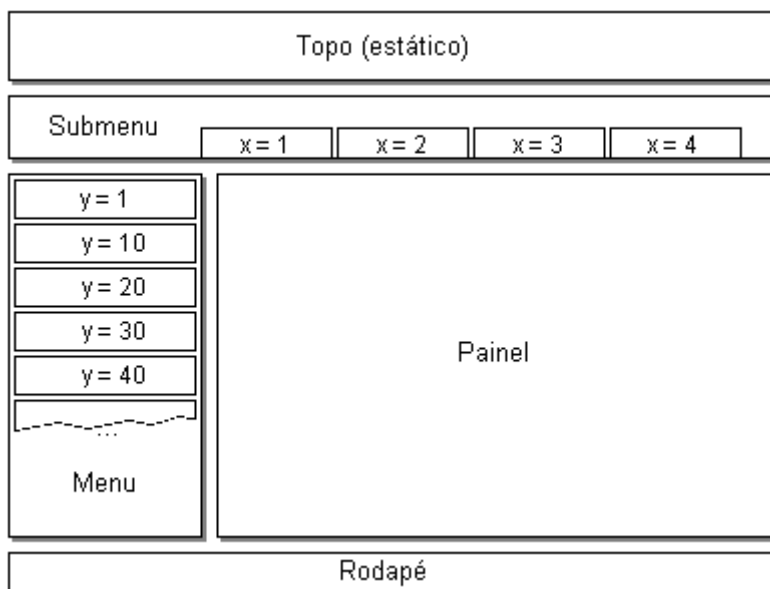


Esquema da Implementação *Command*.

- O *Servlet* instancia um *Vector* de operações no seu método **init()**, controla o fluxo de requisições no seu método **doGet()** e utiliza o método **redireciona()** para fazer o controle do fluxo de respostas.
- A classe abstrata *Operacao* declara o método abstrato, **executar()**, que será implementado nas suas classes filhas conforme o tipo de operação.
- As operações serão armazenadas no *Vector* e chamadas por seu índice e pelo tipo do objeto que fez a requisição da operação.

Utilizaremos o método de envio **post** no formulário de *login* para evitar a exposição de parâmetros na barra de endereços do navegador, enviando a requisição para o método **doPost()** do Servlet Controlador.

A interface do usuário merece comentários adicionais, já que sua implementação será realizada no padrão *Composite*, visando a reutilização de umas poucas páginas e o funcionamento pouco dependente do servidor, rodando quase exclusivamente no cliente. As requisições ao servidor ficarão restritas apenas ao acesso às operações do sistema. O código será escrito em Html, JSP e JavaScript, mas explorando ao máximo a interação entre esses recursos, conforme descrito abaixo.



- O *frame* **Menu** pode emitir mensagens de controle para os outros *frames* e determina um valor *y* conforme a opção selecionada.
- O *frame* **Submenu** também pode emitir mensagens de controle para os demais *frames* e determina um valor *x*, dependente de *y* e da opção selecionada.
- Conforme o par *x,y* enviado para o *frame* **Painel**, uma determinada *tela* será obtida e carregada nesse *frame*, contendo os controles para a operação selecionada.

Frameset da Interface Explodido.

A página JSP Painel, carregará no *frame* Painel, através do seu código Java e dos parâmetros recebidos, um arquivo JS (contendo o código JavaScript controlador do formulário) e um arquivo TXT (Contendo Html e JSP responsáveis pelos controles e operação do formulário) necessários e suficientes para a complementação e montagem da tela selecionada, compondo a página Painel. Assim, a página final obtida e carregada no *frame* Painel, conterá apenas os controles, e códigos de controle, referentes à operação selecionada pelo usuário, sem necessidade de intervenção do Controlador no servidor, além daquela que o JSP requer. A interface possui seu próprio "controlador", a página *interface.jsp*, que se encarrega de receber as respostas do *Servlet* Controlador, "traduzindo-as" para o *frameset*.

Em algumas funcionalidades, quando a tela selecionada pelo usuário, necessitar de algum preparo antes do seu carregamento (como obter dados armazenados na memória previamente), a página *painel.jsp* não será exibida, assumindo o papel de puramente executante e chamando operação adequada para obter o preparo

necessário através do Controlador ao invés de compor e exibir a tela solicitada. A exibição será feita posteriormente, com a tela indicada no método *setProxPagina()* da operação executada.

Os arquivos JS predefinidos ficarão armazenados no pacote *scripts*, enquanto que os arquivos TXT, também predefinidos, ficarão no pacote *includes*. O código JavaScript, além de implementar a ação dos controles do formulário, fará também a crítica do preenchimento dos campos e emitirá mensagens para o usuário.

Não utilizaremos JavaBeans na interface, justificando-se essa opção por três razões: o modelo MVC adotado proverá isolamento adequado e suficiente entre as camadas da aplicação, além disso, não haverá na interface acesso direto a instâncias das classes de modelo, negócio e operações. Finalmente, nosso sistema não utilizará componentes a serem distribuídos.

Na validação do logon e manutenção da sessão do usuário, incluindo suas permissões, nosso código avaliará a instância do objeto Funcionário em lugar da comparação de parâmetros com valores fixos. A permissão é que determinará quais opções estarão disponíveis para o usuário no menu do sistema. Além disso, as operações só poderão ser efetivadas sobre uma permissão validada.

Nossa implementação visará o máximo reaproveitamento de código, principalmente em dois aspectos, conforme descrito a seguir.

Cada classe de Operação (*extend* de *Operacao.java*) será escrita para executar uma das operações do sistema, implementando o método *executar()* da superclasse, para quaisquer dos objetos instanciados. Além dessa função principal, essas classes também implementarão uma "interface tradutora", dentro do método *executar()*, entre os parâmetros reconhecidos na GUI e Servlet (*requests*) e os reconhecidos na DAO (objetos), através do recebimento de parâmetros *request* e o fornecimento de parâmetros Object correspondentes e adequados ao objeto que solicitou a operação.

Na classe DAO, o reaproveitamento se traduzirá também na utilização de *extends*, no sentido de que, cada classe de modelo/negócio possuirá seu próprio DAO para tratar as várias operações e suas respectivas tabelas no BD. Cada uma dessas classes DAO herdará da DAO pai, a conexão com o BD e será acessada por objetos da classe a qual ela corresponde, através de métodos *static* emulando o padrão *Singleton*.



Outra característica do par de classes Operação—DAO será a realização do Mapeamento Objeto—Relacional do nosso sistema, "ajustando" o Modelo de Classes ao Modelo do Banco de Dados e vice-versa, conforme as Regras de Negócio.

Optamos por incluir também nas classes de Modelo e Negócio alguns métodos auxiliares, não listados no Diagrama de Classes, para tratar, calcular, formatar e atribuir valores das propriedades dessas classes, de modo a aproveitar as vantagens da OOP, realizando Regras de Negócio e reduzindo o código nas classes de Operações, de DAO e da interface do usuário.

Sobre a utilização do Banco de Dados, optaremos por criar consultas (*views*) para alimentar as listas de seleção e apresentação de dados e os relatórios do sistema. Isso permitirá manter no BD as consultas (*query*) envolvendo operações de *join* e condições fixas, enquanto o código permanecerá simples e ágil no sistema, fazendo seleções (*selects*) simples com, no máximo, algum filtro ou ordenação por campo. Isso permitirá a obtenção de diversas listas diferentes sobre uma única consulta. Os valores padrão (*default*) dos campos também serão escolhidos para cumprir as Regras de Negócio, evitando, nas inclusões (*includes*), a necessidade de tratar e fornecer valores para todos os campos da tabela envolvida na operação.

Quanto às exclusões, embora os relacionamentos projetados tenham características restritivas para essa operação, implementaremos no código do sistema as verificações necessárias, de modo que o usuário possa ser alertado da impossibilidade da exclusão, quando for o caso, ao invés de deixar que o BD gere um erro de tentativa de violação de restrição.

Para terminar, desejamos destacar que as nossas opções visam resultar em um código de nível profissional, ágil e seguro, mas, tanto o quanto possível, pequeno, simples, de fácil manutenção e que permita, no futuro, a implementação de novas funcionalidades sem a necessidade de grandes alterações.

## **8.1 - POLÍTICA DE LOGIN E SENHAS**

### **8.1.1 – Atribuição do Login e da Senha Inicial**

Ao ser cadastrado, o funcionário receberá como login o seu número de matrícula convertido para texto e a senha padrão “123456”. No primeiro acesso, será solicitada a alteração para uma senha personalizada.

### **8.1.2 – Sigilo, Recuperação e Alteração de Senha**

As senhas serão armazenadas no BD criptografadas e tanto o usuário como o administrador do sistema são responsáveis pela manutenção do sigilo. Não haverá alteração de login no sistema, mas a senha poderá ser alterada a qualquer momento através da opção adequada no menu principal. Caso o usuário esqueça a senha cadastrada, ele poderá solicitá-la na tela de login, através do link “Esqueci minha senha”, recebendo-a no e-mail cadastrado no sistema.

## **8.2 - POLÍTICA DE MANUTENÇÃO E BACKUP**

A manutenção e o backup dos dados serão feitos pelo setor de informática da CISCEA, segundo a política adotada pela empresa, em três linhas de ação:

### **1. Banco de Dados**

Normal diário às 22:00h com histórico de 1 semana.

### **2. Servidor de Arquivos (Aplicações e Arquivos de Documentos da Empresa)**

#### **a. Aplicação**

Normal diário às 22:00h com histórico de 1 semana; Cliente Exchange e Diferencial 22:00h Diário com Histórico de 30 dias.

#### **b. Arquivos de Documentos da Empresa**

Diferencial diário às 22:00h com histórico de 30 dias e Mensal completo (primeira semana do mês) com Histórico de 4 meses (final de semana às 8:00h).

### **3. Arquivos**

Cliente Exchange; Diferencial diário às 22:00h com Histórico de 30 dias.

## CAPÍTULO 9 – MANUAL DO USUÁRIO

O SassWeb foi projetado como uma ferramenta de apoio ao trabalho colaborativo, integrando e somando os esforços de cada área da empresa envolvida na sua utilização, conduzindo e gerindo cada operação nas diversas fases da realização de um atendimento, desde a sua solicitação até a finalização.



A interface amigável, de uso simples, intuitivo e agradável, dispõe de diversos dispositivos informativos para o usuário: dicas de utilização são exibidas no rodapé da página conforme a tela exibida; caixas de diálogo orientam o usuário tanto no preenchimento dos campos como nas condições das operações, alertam sobre o resultado das operações ou permitem ao usuário refletir se realmente deseja realizar determinada operação ou não.

Assim, nosso Manual do Usuário possui um caráter mais aproximado de Guia do Usuário, apenas acrescentando detalhes menos evidentes nas telas da interface, mas perfeitamente adequado a qualquer usuário, principalmente aqueles que já trabalham, ou receberam o treinamento, na empresa.

## 9.1 - TELA PRINCIPAL DE LOGON NO SISTEMA

### 9.1.1 - Logar no Sistema

O logon no sistema é feito com o fornecimento da matrícula e da senha, confirmadas com o botão *Enviar*. Para esvaziar os campos utiliza-se o botão *Limpar*. Através do link *Esqueci Minha Senha* o usuário pode solicitar ao sistema o envio de sua senha para o e-mail cadastrado, mediante a confirmação da sua identidade.

Os dados fornecidos serão validados e se o usuário for um funcionário cadastrado e a senha estiver correta, sua permissão será avaliada. Caso o usuário possua apenas um perfil de acesso, esse perfil será ativado e a entrada no sistema liberada.

Quando o usuário possui mais de um perfil habilitado no sistema, ele deverá selecionar a permissão desejada para a sessão que está iniciando. A escolha deve ser confirmada com o botão *Seguir* para a liberação do acesso ao sistema. Todos os usuários possuem o perfil de Solicitante (Padrão.)

### 9.1.2 - Logout do Sistema

Para encerrar a sessão e sair do sistema basta clicar a opção de menu *Sair*. Uma caixa de diálogo pedirá a confirmação de logout e o usuário poderá confirmar e sair ou cancelar e permanecer na sessão atual.

Se a saída for confirmada a sessão será encerrada e não poderá ser retomada. A sessão de um usuário logado também poderá ser encerrada pelo sistema em caso de erro inesperado ou *timeout* do Servidor Web. Uma vez que a sessão de um usuário estiver encerrada, ele poderá iniciar outra a qualquer momento. O logon com permissões diferentes também é possível, mas em janelas (instâncias) independentes do Navegador Web (*Browser*).

## 9.2 - FUNÇÕES DO PERFIL DE SOLICITANTE

Esse é o perfil Padrão, disponível de modo automático no cadastro do usuário, condicionado apenas à situação (ativo ou inativo) desse funcionário e "embutido" em todos os outros perfis de usuário.

### 9.2.1 - Solicitação de Serviço

Menu *Principal*, submenu *Solicitar Serviço* - Exibe o formulário para Solicitação de Serviços. Todos os campos devem ser preenchidos, sendo que no campo *Detalhes* o usuário explicará, de modo claro e breve, as condições e necessidades do atendimento. O botão *Enviar* emite a solicitação e o botão *Limpar* apaga o conteúdo dos campos. A numeração da solicitação é automática e única para todos os usuários. O campo *Descrição...* exibe um texto explicando o serviço.











Formulário de Solicitação de Serviço:

- Número: 5 (Este número é provisório)
- Área: selecionar (CENTRO DE APOIO)
- Serviço: selecionar (AQUISIÇÃO DE MATERIAIS)
- Descrição do Serviço Selecionado: ESCRITÓRIO E/OU MANUTENÇÃO.
- Detalhes sobre a execução do Serviço: 1 resma papel a4 e 2 canetas (azul e preta)
- Botões: Limpar, Enviar

Duas solicitações do mesmo serviço não poderão ser emitidas ao mesmo tempo por um usuário, ou seja, um serviço A não poderá ser solicitado enquanto uma solicitação do serviço A estiver em andamento.


### 9.2.2 - Consultar Solicitações Pendentes

Menu *Principal*, submenu *Situação Atual* - Apresenta a lista das solicitações emitidas, não canceladas e não encerradas, do usuário. O ícone *Detalhes* (lupa) "expande" a linha e exibe os detalhes da solicitação, enquanto que o ícone *Alterar* (lápis) acessa o formulário para alteração da solicitação. O ícone *Impressora* permite a impressão em arquivo pdf ou em papel da lista exibida na tela.

Situação: <span style="color: green;">■</span> emitida <span style="color: yellow;">■</span> encaminhada <span style="color: red;">■</span> pendente						
(Uma solicitação pode estar Pendente de informação ou de encerramento)						
Solicitações em Andamento						
Número	Serviço	Emitida em	Dependência	Término previsto	Sit	
6	VTI - MANUTENÇÃO DE SISTEMAS	18-12-2008	não tem	22-12-2008	e	 
4	VAP - COLETA DE DOCUMENTAÇÃO SIGILOSA	04-12-2008	não tem	08-12-2008	p	 
3	VTI - INSTALAÇÃO DE SOFTWARE	03-12-2008	não tem	08-12-2008	n	 
2	VDC - CÓPIAS DE ORIGINAIS	01-12-2008	não tem	04-12-2008	p	 
1	VAP - AQUISIÇÃO DE MATERIAIS	29-11-2008	não tem	02-12-2008	n	 

### 9.2.3 - Alterações na Solicitação

Ao clicar o ícone *Alterar* de uma solicitação, será carregado o formulário de alteração preenchido com os dados dessa solicitação. O usuário poderá *Cancelar* a solicitação ou *Informar* mais detalhes. A opção *Encerrar* só será efetiva quando a solicitação estiver atendida. Não há outras alterações possíveis nem exclusão de solicitações.

Solicitação de Serviço No. 2		
Serviço: CÓPIAS DE ORIGINAIS		
Dependência: não tem	Situação: ENCAMINHADA	
Emitida em 01-12-2008	Encaminhada em 01-12-2008	Executada em
<a href="#">Cancelar</a>   <a href="#">Encerrar</a>   <a href="#">Informar</a>		
Informações sobre o serviço 		
Ordem	Data	Informação Solicitada / Fornecida
1	01-12-2008	CONTRATO INFRAERO 2008
2	04-12-2008 04-12-2008	COM OU SEM ANEXOS? COM ANEXOS
referentes a licitações		

Uma solicitação estará *pendente* (**p** vermelho) por duas razões: há pedido de informação adicional não atendido ou a solicitação já foi atendida e aguarda encerramento. No primeiro caso, o usuário deve acessar *Alteração* e responder ao questionamento da solicitação. Ao *informar* o detalhe adicional, a solicitação voltará para o estado *encaminhada* (**n** amarelo). A informação pode ser complementada a qualquer momento. O ícone **e** (verde) indica que a solicitação emitida ainda não foi encaminhada pelo Coordenador para atendimento, ou seja, ainda não consta na Programação de Atendimentos.

### 9.2.4 – Encerramento da Solicitação

Ao selecionar uma solicitação pendente de encerramento, a opção *encerrar* será desbloqueada e, ao ser clicada, exibirá um novo formulário onde o usuário dará nota ao atendimento recebido. Após a valoração, que deverá ser absolutamente imparcial, a solicitação estará encerrada e não aparecerá mais na Lista de Andamento.

Avaliação do Atendimento
Você considera que o atendimento recebido foi...
<div> <div>não avaliado</div> <div> nã avaliado  insatisfatório  normal  bom  muito bom  ótimo </div> </div>
<b>Enviar</b>

Critérios da Consulta	
Por Número	<input type="text"/>
Por Serviço	<b>Todos</b>
Por Situação	<b>Todas</b>
No período de	<input type="text"/> até <input type="text"/>
<b>Enviar</b>	

### 9.2.5 - Consultar Solicitações

Solicitações canceladas e encerradas não aparecem na Lista de Andamento. Então, para consultar uma solicitação em uma dessas situações, acesse o menu

*Solicitações*, submenu *Critérios*. Selecione os critérios desejados e clique o botão *Enviar* para aplicar o filtro de consulta e exibir a listagem do resultado. Se nenhum critério for escolhido, todas as solicitações do usuário serão exibidas.

#### 9.2.6 - Sugestão de Novo Serviço

Menu *Solicitações*, submenu *Incluir Serviço* – Permite o pedido de inclusão de um serviço novo, e deve ser utilizado quando o atendimento desejado não se enquadra em nenhum dos serviços cadastrados. Preencha os dados que deseja sugerir, confirmando com o botão *Enviar*. Seu pedido será avaliado pelo Administrador do sistema.

**Nova Senha Desejada**

Nova Senha

Confirmação

**Gravar**

**Dados Sugeridos para o Novo Serviço Desejado**

Área Sugerida

Título do Serviço

Descrição do Serviço

Prazo padrão  horas (o valor default é 24)

Prioridade

**Enviar**




#### 9.2.7 - Alteração de Senha

Menu *Solicitações*, submenu *Alterar Senha* - Preencha os campos *Nova Senha* e *Confirmação*, depois clique no botão *Gravar*. A senha nova vigorará no próximo login do usuário. No primeiro acesso ao sistema o usuário deve cadastrar sua senha pessoal.

### 9.3 - FUNÇÕES DO PERFIL DE EXECUTANTE

#### 9.3.1 – Consultar os Atendimentos Programados

Menu *Programação*, submenu *Listar* - Exibe a listagem dos Atendimentos Programados que foram encaminhados pelo coordenador para a Equipe do serviço solicitado. Todos os componentes de uma Equipe de Serviço visualizam a mesma lista. O ícone *Alterar* permite operações sobre a solicitação e o ícone *Impressora* imprime o Relatório de Atendimentos Programados.


Atendimentos Programados						
Ordem	Número	Área	Serviço	Solicitante		
1	1	VAP	AQUISIÇÃO DE MATERIAIS	CELIA GEORGINA G DE C MEDEIROS - LIP		
prioridade 2		2 RESMAS PAPEL A4				
Emitida 29-11-2008 Encaminhada 01-12-2008 Prazo 02-12-2008 Executante: FRANCISCO DAS CHAGAS F CABRAL - VAP						

Serão exibidos o detalhe sobre o serviço, o prazo para o término do atendimento e o Executante escalado. Caso esse funcionário não esteja

disponível para realizar o atendimento, um outro membro qualquer da Equipe do Serviço poderá realizá-lo. Os atendimentos relativos a serviços de outra equipe não serão listados.

### 9.3.2 – Alteração e Atendimento da Solicitação

O formulário de Alteração oferece as seguintes opções: *Cancelar* uma solicitação que não poderá ser atendida no momento, *Encerrar* uma solicitação imprópria ou equivocada que tenha escapado à crítica do Coordenador, *Perguntar*, em caso de dúvida, solicitando mais informações ao emitente da solicitação e, finalmente, *Atender* a solicitação cujo serviço já foi executado.

Solicitação de Serviço No. 2			
Serviço: CÓPIAS DE ORIGINAIS		Situação: ENCAMINHADA	
Dependência: não tem		Executada em	
Emitida em 01-12-2008		Encaminhada em 01-12-2008	
		<a href="#">Cancelar</a>	<a href="#">Encerrar</a>
		<a href="#">Perguntar</a>	<a href="#">Atender</a>
Informações sobre o serviço 			
Ordem	Data	Informação Solicitada / Fornecida	
1	01-12-2008	CONTRATO INFRAERO 2008	
2	04-12-2008	COM OU SEM ANEXOS?	
	04-12-2008	COM ANEXOS	
as licitações também?			

## 9.4 - FUNÇÕES DO PERFIL DE COORDENADOR

### 9.4.1 – Relatório de Executantes

Menu *Executantes*, submenu *Critérios* - Carrega o formulário de critérios para o filtro do relatório. Após definir o filtro desejado, clicar o botão *Enviar* para emitir o Relatório de Executantes. Se nenhum critério foi fornecido, todos os executantes serão listados. Nos critérios digitados, podem ser usadas *palavras-chave*, ou parte de um nome, para aumentar a abrangência do filtro. Essa funcionalidade estará disponível, com as mesmas características, no perfil de Administrador.

### 9.4.2 – Relatório de Atendimentos

Idêntico ao anterior, mas focado nos atendimentos realizados. Acessado através do menu *Atendimentos*, também disponibiliza critérios de filtro e impressão do Relatório de Atendimentos. Também faz parte do perfil de Administrador.

### 9.4.3 – Manutenção das Equipes de Serviço

Cada Coordenador terá acesso ao cadastro das Equipes de Serviço pertencentes à sua Área de Coordenação. Essa funcionalidade permite alterar as equipes de cada serviço, incluindo ou removendo Executantes, conforme a necessidade da demanda do serviço ou mesmo as habilidades técnicas dos funcionários.



A funcionalidade é acessada pelo menu *Equipes* e submenu *Alterar*. Um serviço deverá ser selecionado para a apresentação dos membros da Equipe correspondente. Apenas serviços da Área da Coordenação serão listados. O uso dos botões é absolutamente intuitivo e cada alteração realizada é gravada imediatamente. As operações nesse painel são monitoradas pelo sistema.

#### 9.4.4 – Encaminhamento das Solicitações

O menu *Encaminhamento* exibirá uma lista de Solicitações emitidas e ainda não validadas para atendimento. O ícone *Alterar* copia os dados da solicitação para o formulário de *Alteração*, exibindo-os no painel e o ícone *Impressora* permite emitir a lista de Solicitações para Encaminhamento.

Solicitações para Encaminhamento					
Número	Área	Serviço	Emissão	Solicitante	
5	VAP	SERVIÇOS GERAIS	08-12-2008	EDUARDO CESAR DA SILVA LIMA - VDC	
		TROCA DE FACHADURA E CADEADO NA SALA 201			




As alterações permitidas são: *Cancelar* uma solicitação sem condição de atendimento; *Encerrar* uma solicitação equivocada ou imprópria (devolução de solicitação) ou *Encaminhar* o serviço para a programação se tudo parece correto na solicitação. Ao encaminhar, o Coordenador poderá vincular o atendimento, condicionando-o a um outro serviço que seja pré-requisito. Os prazos são atribuídos pelo sistema, já saltando finais de semana. O Coordenador pode acrescentar até 5 dias no prazo, prevendo recessos no funcionamento da empresa. Nesse caso, o sistema verificará se a nova data cai em um final de semana.

A solicitação encaminhada será incluída na Programação de Atendimentos, com um executante sugerido pelo sistema e ordenada, junto às demais, pela data do

encaminhamento, a data final do atendimento (prazo) e a prioridade do serviço. Uma solicitação dependente terá seu prazo normal contado à partir da data final da solicitação pré requisito.

#### 9.4.5 – Programação de Atendimentos

Menu *Programação*, submenu *Listar* – Exibe a Relação de Atendimentos Programados. Nessa lista aparecem todas as solicitações encaminhadas e não executadas, de todas as Áreas, para que cada Coordenador tenha uma visão ampla da situação atual, auxiliando na tomada de decisões. Entretanto, o ícone de *Alteração* estará ativo apenas nas solicitações da sua Área de Coordenação.

Atendimentos Programados						
Ordem	Número	Área	Serviço	Solicitante		
1	1	VAP	AQUISIÇÃO DE MATERIAIS	CELIA GEORGINA G DE C MEDEIROS - LIP		
prioridade 2		2 RESMAS PAPEL A4				
Emitida 29-11-2008 Encaminhada 01-12-2008 Prazo 02-12-2008 Executante: FRANCISCO DAS CHAGAS F CABRAL - VAP						
2	2	VDC	CÓPIAS DE ORIGINAIS	CELIA GEORGINA G DE C MEDEIROS - LIP		
prioridade 3		CONTRATO INFRAERO 2008				
Emitida 01-12-2008 Encaminhada 01-12-2008 Prazo 04-12-2008 Executante: EDUARDO CESAR DA SILVA LIMA - VDC						
3	3	VTI	INSTALAÇÃO DE SOFTWARE	CELIA GEORGINA G DE C MEDEIROS - LIP		
prioridade 4		BR OFFICE 2				
Emitida 03-12-2008 Encaminhada 03-12-2008 Prazo 04-12-2008 Executante: MARCIO SOARES DE LOYOLA - VTI						
4	4	VAP	COLETA DE DOCUMENTAÇÃO SIGILOSA	CELIA GEORGINA G DE C MEDEIROS - LIP		
prioridade 1		DOCUMENTOS PARA O COMAER				
Emitida 04-12-2008 Encaminhada 04-12-2008 Prazo 08-12-2008 Executante: ADILSON CARLOS DE SOUZA - VAP						

O formulário de Alteração possui funções semelhantes ao de Encaminhamento, *Cancelar*, *Encerrar*, *Vincular* (link *Salvar*) e *Postergar* (o prazo, link *Salvar*), exatamente com as mesmas finalidades. As funcionalidades *Vincular* e *Postergar* não podem ser aplicadas simultaneamente. A opção *Adiantar*, permite adiantar a solicitação na programação em um dia ou o necessário para atender os outros critérios de ordenação.

Item 3: Solicitação de Serviço No. 3			
Serviço: INSTALAÇÃO DE SOFTWARE		Emitida em 03-12-2008 Encaminhada em	
Área: VTI			
Solicitante: CELIA GEORGINA G DE C MEDEIROS - DIRETOR - Área LIP			
		<a href="#">Cancelar</a>	<a href="#">Encerrar</a> <a href="#">Salvar</a> <a href="#">Adiantar</a>
Informações sobre o serviço			
Ordem	Data	Informação fornecida	
1	03-12-2008	BR OFFICE 2	
Só faça uma dessas alterações com absoluta segurança e atenção.			
Tornar essa solicitação dependente so atendimento da solicitação no.		3	Novo prazo para atendimento em 0 dias

## 9.5 - FUNÇÕES DO PERFIL DE ADMINISTRADOR

### 9.5.1 – Cadastro de Dados Base

Os dados referentes a Áreas, Cargos, Funções, Funcionários (ou usuários), Serviços e Permissões configuram o rol de Dados Base do sistema, sem os quais não há possibilidade de realizar e acompanhar as Solicitações de Serviço.

Cada um desses cadastros é acessado pela opção de menu correspondente e os submenus apresentados permitem *Incluir*, *Listar*, *Alterar* ou *Excluir* dados. As listagens nessas funcionalidades não possuem critérios de filtro, mas permitem a ordenação por qualquer uma das colunas, bastando clicar o *Título* da coluna desejada para ordenar a listagem. Oportunamente, lembre-se que o Cadastro de Funcionários permite a atribuição de até três permissões acumulativas de perfis de usuário, através de caixas de verificação (*checkbox*). Se nenhuma for marcada, o único perfil do funcionário será o de Solicitante, que não pode ser atribuído ou removido.

Por se tratarem de cadastros normais, não há necessidade de acrescentar mais nada quanto à sua utilização, exceto as orientações constantes no Anexo I.

### 9.5.2 – Manutenção de Coordenações

O menu *Coordenações*, dá acesso ao formulário de Manutenção de Coordenações. Seu funcionamento e utilização é semelhante ao formulário de Manutenção de Equipes (perfil de Coordenador), mas atuando sobre as Coordenações de todos os Serviços disponíveis para solicitação. Uma outra diferença é que, enquanto uma Equipe de Serviço não é limitada em número, cada Coordenação só admite dois Coordenadores, o Principal e o Suplente. A ordem em que são listados na janela *Alocados*, representa exatamente essa hierarquia.

Além da composição das Coordenações, é possível também alterar a hierarquia dos coordenadores, mas essa tarefa deve ser realizada em dois passos: remover o Coordenador Principal e, em seguida, adicioná-lo novamente na Coordenação.

## CAPÍTULO 10 – CONCLUSÕES E TRABALHOS FUTUROS

Como conclusão, destacamos a realização e o sentimento de gratificação experimentados ao satisfazer-mos as expectativas do cliente com a apresentação do protótipo do SassWeb, realização esta, obtida com muito planejamento, trabalho, estudo e aprendizado teórico e prático, além do companheirismo vivenciado no desenvolvimento em equipe. Nossa experiência foi tão positiva que nos sentimos confiantes e animados para formar uma equipe de desenvolvimento e prestação de serviços em projeto de *software*.

Nossos trabalhos futuros envolvem a manutenção do sistema SassWeb e a provável implementação de novos módulos ou novas funcionalidades, entre elas a inclusão de funções estatísticas com gráficos para auxiliar nas estratégias de atendimento da empresa e a utilização de tecnologia J2ME (Java para celular) para atualização da situação em modo *cable-free* ou, pelo menos, enviar alertas e mensagens do sistema através de SMS.



## CAPÍTULO 11 – BIBLIOGRAFIA

ARMSTRONG, Eric [e outros] — The J2EE 1.4 Tutorial, Sun Microsystems inc., Santa Clara, California, 2006.

ALMEIDA, Alexandre de e DAROLT, Reginaldo — Pesquisa e Desenvolvimento em UML. TCC em Ciência da Computação, UNISUL, Santa Catarina, 2001.

BEZERRA, Eduardo — Princípios de Análise e Projeto de Sistemas com UML. Editora Campus, RJ, 3ª Edição.

BRAZ, Cristian Cleber Masdeval — Principais Padrões J2EE para Construção de Aplicações Não Distribuídas, Tutorial Dataprev—GUJ.

CAVALCANTE, Eduardo Henrique e MOURA, Pedro N. de Souza — Implementação de Sistemas. TCC do Bacharelado em Ciência da Computação, UFRJ, 2007.

COOPER, James W. — The Design Patterns Java. Addison-Wesley, USA, 1998.

DEITEL, H. M. e P. J. Deitel. — Java, Como Programar. Porto Alegre, Bookman, 2003, 4ª Edição.

FALKNER, Jayson & JONES, Kevin — Servlets and Java Server Pages: The J2EE Technology Web Tier. Addison-Wesley, Pearson Education Inc., Boston, USA, 2004.

HEUSER, Carlos Alberto — Projeto de Banco de Dados. Instituto de Informática da UFRGS, Editora Sagra, 4ª Edição, 1998.

HOLZNER, Steve — Eclipse — A Java Developers Guide. O'Reilly Media Inc., Sebastopol, USA, 2004.

MARTIN, Robert Cecil — UML for Java Programmers. Prentice Hall, USA, 2002.

MYSQL AB — MySQL 5.0 Reference Manual. MySQL Community Server, 2008.

NAIBURG, Eric J. & MACKSINCHUK, Robert A. — UML for Database Design. Addison-Wesley, USA, 2001, 1<sup>st</sup> Edition.

PERRY, Bruce W. — Java Servlet & JSP Cookbook. O'Reilly Media Inc., Sebastopol, USA, 2004.

REESE, George — Database Programming with JDBC and JAVA. O'Reilly, USA, 2001, 2<sup>nd</sup> Edition.

SCHMULLER, Joseph — Aprendendo UML em 24 Horas. Prentice Hall, Pearson Educação Latinoamérica, México.

UMRYCH, Cary E. & AHMED, Khawar Zaman — Developing Enterprise Java Applications with J2EE and UML. Addison—Wesley, USA, 2001, 1<sup>st</sup> Edition.

UNIVERCIDADE – Centro Universitário da Cidade do Rio de Janeiro — Normas para o Projeto Final de Curso: Elaboração, Apresentação da Monografia e Avaliação da Disciplina “Trabalho de Conclusão de Curso”. Rio de Janeiro, 2008.

## 11.1 – LEITURAS ADICIONAIS

BARBALHO, Sanderson C. Macêdo; ROZENFELD, Henrique; AMARAL, Daniel Capaldo — Modelando Processos de Negócio com UML, XXII Encontro Nacional de Engenharia de Produção, Curitiba, PR, 2002.

ROCHA, Helder da — Padrões de Projeto J2EE - Padrões da camada de apresentação (WEB). Material do curso J931: J2EE Design Patterns, Argonavis, 2003.

SALM JUNIOR, José Francisco — Extensões da UML para Descrever Processos de Negócio, Dissertação de Pós-Graduação da UFSC, Florianópolis, SC, 2003.

SORROCHE, Rogério e LOPES, Maurício Capobianco — Uso de Design Patterns e J2EE: um estudo de caso. Trabalho da FURB, sem data.

## ANEXO I

### 1. OBJETIVOS DA CONSCIENTIZAÇÃO NO USO DO SISTEMA

Formar nos funcionários da empresa um comportamento de compromisso colaborativo em relação ao uso do sistema, de modo a obter o máximo rendimento dessa ferramenta de gestão. Para tanto, espera-se dos usuários as atitudes listadas a seguir.

#### 1.1. Perfil de Solicitante (todos os usuários)

- Sempre utilizar o sistema para fazer solicitações, evitando sob qualquer pretexto, os pedidos pessoais ou por memorando.
- Expor com clareza e precisão os detalhes sobre as necessidades e condições do serviço solicitado.
- Atentar para a área e serviço selecionados no tocante ao tipo de atendimento desejado, evitando o cancelamento de solicitações equivocadas.
- Evitar solicitações apressadas, desnecessárias ou que já tenham sido emitidas por um colega de seção, mas fazer a solicitação imediatamente após a constatação de sua necessidade real.
- Verificar a situação das solicitações emitidas diariamente para fornecer mais detalhes sobre o serviço quando solicitado.
- Avaliar e encerrar as solicitações logo após sua execução, focando a avaliação no *atendimento* e não no *atendente*, relevando diferenças e simpatias pessoais.
- Manter sigilo absoluto sobre sua senha de acesso.

#### 1.2. Perfil de Executante

- Respeitar o encaminhamento do serviço. Em caso de necessidade um executante não escalado da equipe poderá realizar o serviço e dar baixa no atendimento.
- Respeitar a ordem dos atendimentos constantes na programação.
- Solicitar mais informações nos casos em que houver dúvida sobre as condições do serviço solicitado.
- Atualizar a situação das solicitações imediatamente após os atendimentos.

#### 1.3. Perfil de Coordenador

- Manter ao menos um executante por equipe nos serviços ativos.
- Evitar o acúmulo de solicitações emitidas e não encaminhadas.
- Adiantar na ordem da programação apenas os atendimentos de real urgência.
- Aplicar com atenção a postergação de atendimentos em virtude de recessos no funcionamento da empresa ou vinculação a outro atendimento.

#### 1.4. Perfil de Administrador

- Cadastrar os novos funcionários no sistema tão logo sejam admitidos.
- Avaliar e atender com presteza os pedidos de inclusão de novos serviços.
- Manter ao menos um coordenador na coordenação de cada serviço ativo.
- Manter contato com o pessoal do SGDB da empresa para eventuais alterações nos cadastros de dados-base do sistema.

## ANEXO II

## MAPA DAS OPERAÇÕES DO SISTEMA SASSWEB — BLOCO 1

	TxtObj	0	1	2	3	4
<b>operacao</b>		Area	Cargo	Funcao	Funcionario	Servico
TELA		402	412	422	432	442
<b>0</b>		listar <sup>1</sup>	listar <sup>1</sup>	listar <sup>1</sup>	listar <sup>1</sup>	listar <sup>1</sup>
TELA						23
<b>1</b>						novo
TELA						—
<b>2</b>						exibir (para solicitar)
TELA		401	411	421	431	441
<b>3</b>		incluir	incluir	incluir	incluir	incluir
TELA		403	413	423	433	443
<b>4</b>		alterar	alterar	alterar	alterar	alterar
TELA		404	414	424	434	444
<b>5</b>		excluir	excluir	excluir	excluir	excluir
TELA						
<b>6</b>						
TELA						
<b>7</b>						
TELA						21
<b>8</b>						validar (srv solicit)
TELA						242
<b>9</b>						trocar (prazos atd)

Obs.: 1. Os itens assinalados são carregados em listas diretamente via DAO para alimentar as caixas de listagem na Interface do Usuário.



## MAPA DAS OPERAÇÕES DO SISTEMA SASSWEB — BLOCO 2

	TxtObj	5	6	7	8	9
<b>operacao</b>		Administrador	Permissao	Coordenador	Executante	Solicitacao (*)
TELA		—	462	—		12
<b>0</b>		listar coordenador	listar <sup>1</sup>	listar executante		listar (pendentes + detalhes)
TELA				232		11
<b>1</b>				novo encaminhamento de sol		novo (registro)
TELA		201-2 / 211-12		231		22
<b>2</b>		exibir <sup>3</sup> rel exec / rel atend		exibir sol emitida		exibir
TELA			461	232	102	11
<b>3</b>			incluir	incluir encam sol na programação	incluir (pedir detalhe)	incluir
TELA		451	463	221	102	13
<b>4</b>		alterar coordenação	alterar	alterar equipe	alterar (atendida)	alterar (detalhe)
TELA			464			
<b>5</b>			excluir			
TELA				232-242	102	13
<b>6</b>				cancelar solicitação	cancelar solicitação	cancelar
TELA				232-242	102	14
<b>7</b>				encerrar solicitação	encerrar solicitação	encerrar
TELA		211		241-242	101-102	logon
<b>8</b>		validar <sup>3</sup> crit rel atend		validar programação	validar programação	validar <sup>2</sup>
TELA				242		24
<b>9</b>				trocar (ordem sol)		trocar <sup>2</sup> (senha)

Obs.: (\*) 2. O objeto 9 refere-se às Solicitações ou aos Solicitantes (usuários) onde assinalado.

3. Atende também ao perfil de Coordenador.

## ANEXO III

### **PAPEL DE TABELAS, RELACIONAMENTOS E CONSULTAS NA REALIZAÇÃO DAS REGRAS DE NEGÓCIO CONFORME O MODELO OO**

Obs.: Consultar o Diagrama de Classes, pág. 38, e o Projeto Lógico do BD, pág. 50.

#### 1. Classes Funcionario, Solicitante, Executante, Coordenador e Administrador

Os quatro perfis possíveis são, obrigatoriamente, objetos Funcionario. Por outro lado, o perfil Solicitante (funcionário logado) está contido nos três perfis restantes, que não são exclusivos, no sentido de que podem ser acumulados por um mesmo funcionário. As tabelas Funcionarios, Usuarios e Permissoes realizam esse cenário através dos seus relacionamentos.

As consultas aplicadas sobre essas tabelas produzem os seguintes resultados:

- *view* ListaFuncionarios..... Dados dos funcionários
- *view* ListaUsuarios ..... Permissões atribuídas a cada funcionário

#### 2. Classes Servico, Coordenador e Executante

Cada serviço possui uma coordenação composta por nenhum (serviço inativo), um ou dois funcionários coordenadores pertencentes à mesma área desse serviço. Como há um número máximo definido, a tabela Coordenacoes sempre conterá dois registros para cada serviço preenchidos com código zerado. As atribuições ou remoções de coordenadores sempre implicarão em *update*, atualizando os registros para o código do funcionário ou para zero, conforme o caso.

De modo semelhante, cada serviço possui uma equipe de executantes, entretanto, nesse caso, não há um limite máximo de membros na equipe. Assim, na tabela Equipes haverá inclusão e/ou exclusão, prevendo-se uma renumeração da ordem de entrada dos funcionários, mantendo-a sem lacunas.

As tabelas Coordenacoes e Equipes, ambas relacionadas com Servicos, permitem obter, através de consultas, as seguintes listagens:

- *view* AreaCoordenador..... Áreas dos coordenadores
- *view* ConsultaExecutantes ..... Dados dos executantes dos serviços
- *view* ListaCoordenacoes..... Dados das coordenações dos serviços
- *view* ListaCoordenadores ..... Os funcionários com essa função
- *view* ListaEquipes..... Dados das equipes dos serviços
- *view* ListaExecutantes ..... Os funcionários com essa função
- *view* ServicoEquipe..... Serviços das equipes

#### 3. Classe Solicitacao

O atendimento de um serviço solicitado pode depender do atendimento prévio de um outro serviço constante em outra solicitação. Assim, assumimos que toda solicitação depende de outra (pré-requisito) e, caso isso não seja verdade, ela depende de si própria. O auto-relacionamento na tabela Solicitacoes realiza esse cenário, enquanto que a *view* ConsultaSolicitações fornece os dados das solicitações, incluindo suas dependências.

#### 4. Classes Solicitacao, Detalhe e Programacao

Toda solicitação é emitida pelo solicitante incluindo um detalhe sobre o serviço, ou seja, sobre o serviço padronizado, qual a tarefa desejada, suas condições, restrições, etc. O executante poderá pedir mais detalhes sobre essas condições de realização do serviço e uma informação complementar será fornecida pelo solicitante. Esse "bate-papo" ficará registrado na tabela Detalhe.

O critério para o atendimento das solicitações validadas e encaminhadas (incluídas na programação de atendimentos) é, nessa ordem, a data da emissão, o prazo para execução (data prevista para o término) e, por fim, a prioridade do serviço. De modo a evitar muita movimentação na tabela Solicitacoes, a ordem do atendimento é armazenada na tabela Programacao, permitindo mobilidade e reordenação com facilidade.

As *views* associadas a essas tabelas permitem listagens e alterações, conforme indicado:

- *view* ConsultaDetalhes ..... Lista os detalhes das solicitações
- *view* ConsultaProgramacao ..... Dados das solicitações programadas
- *view* ListaSolicitados ..... Lista os serviços solicitados
- *view* OrdemAtendimento ..... Lista as solicitações e seus pré-requisitos por ordem e prazo de atendimento

#### 5. Classes Solicitacao, Solicitante e Funcionario

Cada solicitação envolve cinco funcionários em diferentes momentos. Na emissão o solicitante é obrigatório, mas os outros 4 não são conhecidos ainda. Ao ser encaminhada, a solicitação recebe o coordenador que fez o encaminhamento e o executante sugerido pelo sistema (aquele com menos atribuições de atendimento na equipe do serviço). Quando a solicitação é atendida, fica registrado o executante efetivo, que pode ou não, ser o mesmo que estava programado. Finalmente a solicitação recebe o encerrante, o que acontecerá sob quatro cenários:

- encerrada pelo solicitante (é exigida a valoração do atendimento recebido)
- cancelada pelo solicitante (o motivo da solicitação cessou)
- cancelada pelo coordenador/executante (a solicitação não pode ser atendida no momento)
- encerrada pelo coordenador/executante (a solicitação é incoerente)

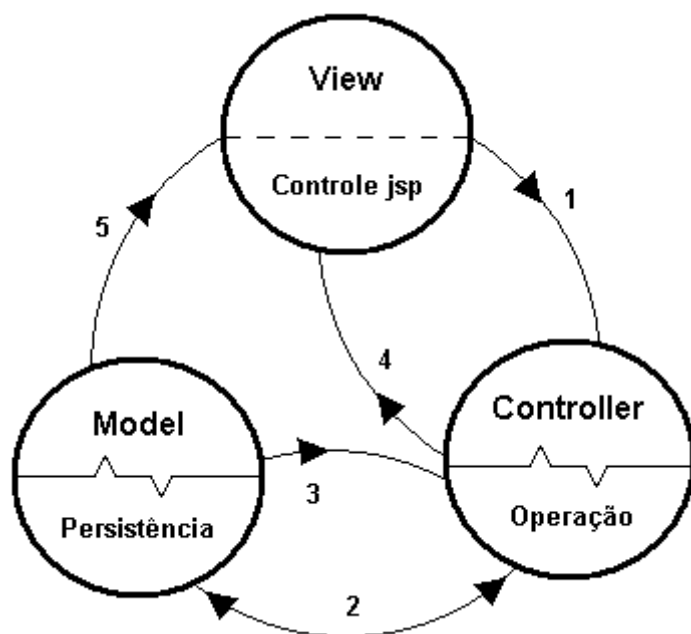
As *views* que auxiliam e sustentam essas operações com listagens e manipulações são:

- *view* ContaAtendimentos ..... Fornece a quantidade de atendimentos dos executantes
- *view* ListaAtendimentos ..... Dados sobre os atendimentos programados



### 3. TROCA DE MENSAGENS DURANTE OS CICLOS

Entendemos por *mensagem*, todo conteúdo útil no fluxo, sejam valores de variáveis, mensagens para o usuário, *ArrayList* de objetos mantidos na sessão, metadados, etc. O Modelo implementado, e o ciclo por extensão, possui pontos de *interfaceamento* (ou "tradução") dessas mensagens, distribuídos entre as suas camadas.



- O controlador da Interface (View) utiliza suas variáveis de controle para decidir o que será exibido, conforme os valores obtidos ou recebidos.
- Da Interface para o Controlador (Controller) (1) são enviadas as mensagens HTTP para o Servlet que ativa a Operação solicitada, transferindo os dados.
- Entre a Operação, o Modelo e a DAO (2, 3), as mensagens são traduzidas do HTTP para objetos do Modelo, ou traduzidas desses objetos para HTTP na forma de *Arrays* de objetos na sessão, e os *Arrays* enviados como resposta para o Controlador.
- O retorno do Controller para a Interface (4) é definido por dois métodos das classes de Operação, *setProxPagina()* e *getProxPagina()*. Isso é importante, porque é dessa forma que o retorno ocorre através do *Servlet* Controlador pelo seu método *redireciona()*.
- É a comunicação entre a Operação, o Modelo e a DAO (Persistência) que sustenta o Mapeamento Objeto-Relacional do sistema, funcionando como um tradutor entre o Modelo Orientado a Objetos e o Banco de Dados Relacional.
- Para que a Interface possa "entender" os objetos recebidos nos *Arrays* da sessão, a comunicação (5) com Model fornecerá informações sobre as classes do Modelo. Embora, como foi explicado neste trabalho, a Interface nunca instancie objetos do Modelo, o acesso às suas propriedades é fundamental para que os valores recebidos possam ser lidos dos *Arrays* através de *casting*.

Por fim, sobre a decodificação nas classes de operação, um artifício foi necessário para reduzir com simplicidade a quantidade dessas classes. Como cada uma é responsável por uma única operação, definida pelo índice do *Vector* de operações *vOp* e a variável *operacao*, então ela deverá realizar essa operação para objetos diferentes ou apresentar uma versão para cada objeto do Modelo. De acordo com a proposta de simplicidade e economia, foi adotada a variável *txtObj* que, em conjunto com a variável *operacao*, determina uma operação específica sobre um objeto específico, conforme o mapeamento apresentado no Anexo II, mantendo um número mínimo de classes de operação.

