



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

DANIEL SOUZA DE JESUS

DESENVOLVIMENTO DE UM SISTEMA INTEGRADO DE GESTÃO
PARA OUVIDORIA

Assis

2013

DANIEL SOUZA DE JESUS

DESENVOLVIMENTO DE UM SISTEMA INTEGRADO DE GESTÃO
PARA OUVIDORIA

Trabalho de Conclusão de
Curso apresentado ao Instituto
Municipal de Ensino Superior de
Assis Como requisito para obtenção o
grau de Tecnólogo em Análise e
Desenvolvimento de Sistemas

Orientador: Me. Douglas Sanches da Cunha
Área de Concentração: Desenvolvimento de Sistemas

Assis
2013

FICHA CATALOGRÁFICA

JESUS, Daniel Souza de

Desenvolvimento de um Sistema Integrado de Gestão para Ouvidoria / Daniel Souza de Jesus. Fundação Educacional do Município de Assis – Assis, 2013.

85 p.

Orientador: Me. Douglas Sanches da Cunha.

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

1. Programação C# em Camadas
2. Sistemas de Informação
3. Ouvidoria
4. Desenvolvimento de Software
5. Plataforma .NET
6. Análise de Sistemas

CDD: 001.61
Biblioteca da FEMA

DESENVOLVIMENTO DE UM SISTEMA INTEGRADO DE GESTÃO PARA OUVIDORIA

DANIEL SOUZA DE JESUS

Trabalho de Conclusão de
Curso apresentado ao Instituto
Municipal de Ensino Superior de
Assis Como requisito para obtenção o
grau de Tecnólogo em Análise e
Desenvolvimento de Sistemas

Orientador: Me. Douglas Sanches da Cunha

Analisador: Esp. Célio Desiró

Assis
2013

DEDICATÓRIA

Dedico este trabalho a Deus em primeiro lugar,
a minha mãe Eunice de Souza,
a toda minha família,
aos meus amigos e professores.

AGRADECIMENTOS

Agradeço primeiramente a Deus por ter me sustentado durante esta fase da minha vida, e por ter me dado força, coragem e sabedoria para enfrentar as dificuldades ao longo do caminho.

Ao professor Douglas pela orientação e pelo constante estímulo transmitido durante o desenvolvimento do trabalho, a todos os professores que me incentivaram a continuar, me ensinaram coisas importantes também sobre a vida.

Aos meus amigos que de alguma forma me ajudaram compartilhando seus conhecimentos e incentivando a continuar, em todos os momentos ao meu lado.

Aos meus familiares que durante todo o percorrer do curso estiveram ao meu lado, em especial minha mãe Eunice de Souza que sempre me estimulou a estudar, mostrando a importância que isso poderia ter em minha vida.

Agradeço a todos que diretamente ou indiretamente me ajudaram para que este trabalho se concretizasse.

RESUMO

Este trabalho apresenta uma análise das necessidades de implementar um Sistema de Ouvidoria na cidade de Assis. Com base em tal análise, foi desenvolvido um sistema online para registrar qualquer tipo de manifestação que um cidadão queira realizar sobre os serviços oferecidos pelo município, podendo registrar também elogios e sugestões para que haja melhora desses serviços prestados.

O trabalho também discute sobre a viabilidade da informatização desse tipo manifestação, mostrando as vantagens em relação a outros tipos de comunicação que ainda são utilizados em pequenos municípios.

Os conceitos teóricos são apresentados de forma resumida e simplificada, para que todos os tipos de leitores se familiarizem com tais conceitos, e que possam compreender a necessidade e a importância que o projeto oferece para as pessoas.

Palavras-chave: 1. Programação C# em Camadas 2. Sistemas de Informação 3. Ouvidoria 4. Desenvolvimento de Software 5. Plataforma .NET 6. Análise de Sistemas

ABSTRACT

This work presents an analysis of the needs to implement a System of Ombudsman in the city of Assis. Based on this analysis, we developed an online system to register any type of event that a citizen wants to carry on the services offered by the municipality, and may also record praise and suggestions for improvement in these services there.

The paper also discusses the feasibility of computerization such demonstration, showing the advantages over other types of communication that are still used in small municipalities.

The theoretical concepts are presented in summary form and simplified so that all types of readers to become familiar with these concepts, and they understand the need and importance of the job offers to people.

Keywords: 1. Programming in C # Layers 2. Information Systems 3. Ombudsman 4. Software Development 5. Platform. NET 6. Systems Analysis

LISTA DE ILUSTRAÇÕES

Figura 1 – Exemplo do fluxo de um sistema.....	25
Figura 2 – Linguagens de programação mais usadas em 2012.....	41
Figura 3 – Mapa mental do Sistema de Ouvidoria.....	47
Figura 4 – Representação gráfica de um ator do diagrama de casos de uso.....	49
Figura 5 – Representação gráfica do relacionamento entre ator e caso de uso.....	50
Figura 6 – Diagrama de casos de uso – Sistema de Ouvidoria.....	51
Figura 7 – Caso de uso realizar manifestação.....	53
Figura 8 – Caso de uso acompanhar manifestação.....	53
Figura 9 – Caso de uso encaminhar manifestação.....	54
Figura 10 – Caso de uso analisar manifestação pendente.....	54
Figura 11 – Caso de uso responder manifestação.....	55
Figura 12 – Caso de uso gerar relatório.....	55
Figura 13 – Caso de uso analisar pendentes.....	56
Figura 14 – Caso de uso responder manifestação (setor).....	56
Figura 15 – Caso de uso manter usuários.....	57
Figura 16 – Caso de uso manter assunto.....	57
Figura 17 – Caso de uso manter setor.....	57
Figura 18 – Caso de uso manter Manifestante.....	58
Figura 19 – Caso de uso manter Ouvidor.....	58
Figura 20 – Caso de uso manter Setor Responsável.....	59
Figura 21 – Diagrama de classes do Sistema de Ouvidoria.....	60
Figura 22 – Diagrama de entidade e relacionamento do Sistema de Ouvidoria.....	61
Figura 23 – Diagrama de sequência realizar manifestação.....	62
Figura 24 – Diagrama de sequência acompanhar manifestação.....	63
Figura 25 – Diagrama de sequência encaminhar manifestação.....	64
Figura 26 – Diagrama de sequência analisar manifestação pendente(ouvidor).....	65
Figura 27 – Diagrama de sequência responder manifestação (ouvidor).....	66
Figura 28 – Diagrama de sequência gerar relatórios.....	67

Figura 29 – Diagrama de sequência analisar manifestação pendente.....	68
Figura 30 – Diagrama de sequência responder manifestação (setor responsável)...	69
Figura 31 – Diagrama de sequência manter usuários.....	70
Figura 32 – Diagrama de sequência manter assunto.....	70
Figura 33 – Diagrama de sequência realizar setor.....	71
Figura 34 – Diagrama de sequência manter manifestante.....	72
Figura 35 – Diagrama de sequência manter ouvidor.....	72
Figura 36 – Diagrama de sequência manter setor responsável.....	73
Figura 37 – Diagrama de atividade realizar manifestação.....	73
Figura 38 – Pasta de camadas do projeto e seus arquivos.....	75
Figura 39 – Tela de início de Sistema de Ouvidoria.....	77
Figura 40 – Tela de login do sistema.....	78
Figura 41 – Tela manter usuário.....	79
Figura 42 – Tela registrar manifestação.....	80
Figura 43 – Tela acompanhar manifestação.....	81
Figura 44 – Tela status da manifestação.....	82

LISTA DE TABELAS

Tabela 1 – Lista de eventos.....	48
Tabela 2 - Cronograma.....	74

SUMARIO

CAPÍTULO 1 - INTRODUÇÃO	16
1.1 JUSTIFICATIVAS	17
1.2 - DÚVIDAS COM RELAÇÃO À VIABILIDADE DO PROJETO.....	18
1.3 – OBJETIVOS A SEREM ALCANÇADOS	18
1.4 – PÚBLICO ALVO	19
1.5 MOTIVAÇÃO DO PROJETO.....	19
CAPÍTULO 2 – CONCEITOS TEÓRICOS.....	21
2.1 - OUVIDORIA	21
2.2 – DEFININDO SISTEMA	22
2.3 - INFORMAÇÃO	23
2.4 – SISTEMAS DE INFORMAÇÃO	24
2.5 – ANÁLISE DE SISTEMAS	26
2.6 – ENGENHARIA DE SOFTWARE	27
2.7 – UML	29
2.7.1 – Diagrama de Caso de Uso	30
2.7.2 – Diagrama de Atividades.....	31
2.7.3 – Diagrama de Classes	32
2.7.4 – Diagrama de Sequência.....	33
2.8 – ORIENTAÇÃO A OBJETOS.....	33
2.8.1 – Classe.....	35
2.8.2 – Atributo	35
2.8.3 – Método.....	36

2.8.4 – Objeto	36
2.9 – BANCO DE DADOS	37
2.10 – LINGUAGEM DE PROGRAMAÇÃO	39
CAPÍTULO 3 - MÉTODOS DE DESENVOLVIMENTO E FERRAMENTAS	42
3.1 – C# (C SHARP).....	42
3.2 – FRAMEWORK .NET	43
3.3 – VISUAL STUDIO	43
3.4 – ASTAH.....	43
3.5 – SQL SERVER.....	44
3.6 – SQL MANAGEMENT STUDIO	44
3.7 – SERVIDOR IIS.....	44
CAPÍTULO 4 – ANÁLISE DO SISTEMA.....	46
4.1 – DADOS SOBRE O CLIENTE	46
4.2 – LEVANTAMENTO DE REQUISITOS	46
CAPÍTULO 5 – DIAGRAMAS	49
5.1 – DIAGRAMA DE CASOS DE USO – DEFINIÇÃO	49
5.1.1 – Diagrama de casos de uso – Aplicação	51
5.2 – DESCRIÇÕES DOS CASOS DE USO DO SISTEMA	52
5.2.1 – Realizar manifestação.....	52
5.2.2 – Acompanhar manifestação.....	53
5.2.3 – Encaminhar manifestação	53
5.2.4 – Analisar manifestação pendente (Ouvidor)	54
5.2.5 – Responder manifestação (Ouvidor).....	54

5.2.6 – Gerar relatórios	55
5.2.7 – Analisar pendentes	55
5.2.8 – Responder manifestação (Setor Responsável)	56
5.2.9 – Manter usuários.....	56
5.2.10 – Manter assunto	57
5.2.11 – Manter setor	57
5.2.12 – Manter Manifestante.....	58
5.2.13 – Manter Ouvidor	58
5.2.14 – Manter Setor Responsável	58
5.3 – DIAGRAMA DE CLASSES – DEFINIÇÃO	59
5.4 – DIAGRAMA DE ENTIDADE E RELACIONAMENTO – DEFINIÇÃO	61
5.5 – DIAGRAMAS DE SEQUÊNCIA.....	62
5.5.1 – Diagrama de sequência – Realizar manifestação	62
5.5.2 – Diagrama de sequência – Acompanhar manifestação.....	62
5.5.3 – Diagrama de sequência – Encaminhar manifestação	64
5.5.4 – Diagrama de sequência - Analisar manifestação pendente (ouvidor)	64
5.5.5 – Diagrama de sequência – Responder manifestação (ouvidor)	66
5.5.6 – Diagrama de sequência – Gerar relatórios.....	67
5.5.7 – Diagrama de sequência – Analisar manifestações pendentes (setor responsável)	68
5.5.8 – Diagrama de sequência –Responder manifestação (setor responsável)	69
5.5.9 – Diagrama de sequência – Manter usuários.....	70
5.5.10 – Diagrama de sequência – Manter assunto	70
5.5.11 – Diagrama de sequência – Manter setor	71

5.5.12 – Diagrama de sequência – Manter manifestante.....	72
5.5.13 – Diagrama de sequência – Manter ouvidor.....	72
5.5.14 – Diagrama de sequência – manter setor responsável.....	73
5.6 – DIAGRAMA DE ATIVIDADE – REALIZAR MANIFESTAÇÃO	73
5.7 – CRONOGRAMA	74
CAPÍTULO 6 – IMPLEMENTAÇÃO.....	75
6.1 – ORGANIZAÇÃO DO PROJETO.....	75
6.2 – INTERFACE	76
6.2.1 – Tela de início.....	77
6.2.2 – Tela de login	78
6.2.3 – Tela manter usuário	79
6.2.4 – Tela registrar manifestação	79
6.2.5 – Tela acompanhar manifestação	80
6.2.6 – Tela status da manifestação.....	82
CONSIDERAÇÕES FINAIS.....	83
REFERÊNCIAS BIBLIOGRÁFICAS	84

CAPÍTULO 1 - INTRODUÇÃO

Com o grande crescimento do fluxo de informações e o aumento da necessidade das empresas se relacionarem com seus clientes e cidadãos, a aplicação da tecnologia nas tomadas de decisões está se tornando cada vez mais necessária, principalmente pelo fato dessas informações ficarem mais organizadas, e também pelo retorno que empresa terá em relação aos dados que ela obterá através de um sistema informatizado, onde o foco principal é a satisfação de ambas as partes.

Com o passar dos anos, as pessoas estão ficando cada vez mais exigentes em relação a tudo que relaciona a tempo e dinheiro, devido ao grande estresse do dia a dia, consequentes de seus problemas de relacionamento e insatisfações com alguns tipos de serviços requisitados no seu cotidiano. Devido a esse fato é que surge a necessidade dos órgãos públicos e até mesmo empresas privadas adotarem um Sistema de Ouvidoria para se relacionarem e resolverem seus consecutivos problemas, podendo então registrar reclamações, elogios, denúncias, sugestões, que será registrado através de uma manifestação.

Outro aspecto muito importante que as grandes empresas estão buscando no mercado, é o baixo custo na implementação de seus sistemas, porém com um sistema de alta qualidade e com um banco de dados seguro e eficiente.

No primeiro capítulo são abordadas as justificativas do sistema, ou seja, por que é necessário criar um Sistema de Ouvidoria, as dúvidas relativas à viabilidade do projeto, os objetivos e o público alvo ao qual o sistema será destinado, as motivações do projeto e as ferramentas utilizadas para a análise do projeto e desenvolvimento e documentação do sistema.

Os conceitos teóricos são apresentados no segundo capítulo, com o objetivo de apresentar e descrever esses conceitos de uma maneira breve e clara, de maneira que os leitores que não estão inseridos no meio da Análise e Desenvolvimento de Sistemas venham compreender o seu contexto, sendo então importante

compreender o que é um sistema, o que é caso de uso e suas especificações, UML, entre outros tópicos abordados.

O terceiro capítulo é onde são discutidos os métodos utilizados para o desenvolvimento do sistema, dando foco as ferramentas usadas para documentação, análise e o desenvolvimento do software, e também o trabalho escrito. As ferramentas Visual Studio, SQL Server, Astah, SQL Management Studio e o servidor IIS serão brevemente descritas e o seu uso é também justificado.

O quarto capítulo é uma breve análise do cliente do sistema, cujos seus dados são apresentados conforme a necessidade e privilégio de acesso ao sistema, englobando os requisitos desejados pelo mesmo.

No quinto capítulo está sendo apresentados os diagramas de Caso de Uso, Diagramas de Classe e de Entidade e Relacionamento, onde torna possível o leitor visualizar todo o funcionamento do sistema após sua finalização. Após esses capítulos, é apresentada uma breve conclusão, relatando como foi o desenvolvimento do sistema, seus êxitos quanto seus fracassos.

1.1 JUSTIFICATIVAS

A vantagem principal para a utilização do sistema integrado em um órgão público, não é apenas o fato de se estar informatizando esse tipo de serviços, mas sim uma estrutura organizacional totalmente flexível e contínua, pois a consolidação do órgão com o cidadão é o aspecto mais importante em um Sistema Ouvidoria.

Um órgão ou empresa ao apresentar um Sistema de Gestão Integrado de Ouvidoria, estará mostrando que tem um compromisso e preocupação com seus clientes, deixando um pouco de lado a perspectiva do crescimento patrimonial e demonstrando sua preocupação com a satisfação das pessoas que desfrutam de seus serviços.

1.2 - DÚVIDAS COM RELAÇÃO À VIABILIDADE DO PROJETO

As perguntas mais latentes do projeto são as seguintes:

- É possível criar um sistema flexível que atenda todas as necessidades dos usuários, podendo registrar todos os tipos de manifestações possíveis atendendo as satisfações do mesmo?
- Os cidadãos terão interesse e confiabilidade em usar esse tipo de serviço para registrar suas manifestações perante um setor público, sem receio em fornecer seus dados mesmo sabendo que serão usados apenas para o acompanhamento de seu próprio cadastro?
- Ao ser implementado em uma estrutura organizacional onde não haja esse tipo de serviço, a imagem desse órgão público ou empresa perante seus clientes continuará sendo a mesma em relação à satisfação dos serviços prestados?

1.3 – OBJETIVOS A SEREM ALCANÇADOS

O trabalho tem como objetivo desenvolver um Sistema Integrado de Gestão de Ouvidoria, desenvolvido na plataforma ASP.NET E utilizando a linguagem C# para programação, tornando assim dinâmica a acessibilidade que o ambiente fornece ao usuário .

O sistema terá a capacidade de registrar as manifestações geradas por um cidadão quando aplicado em um órgão público. Essas manifestações poderão ser feitas através de reclamações, elogios, denúncias, sugestões ou qualquer outro fator que possibilite um melhor relacionamento entre o órgão e o cidadão, minimizando os problemas decorridos da má qualidade no atendimento e prestação de serviços em geral.

A partir das observações feitas do funcionamento de Sistemas de Ouvidorias em vários tipos de empresas e órgãos públicos, e também analisando a situação da cidade de Assis na atual conjuntura, percebe-se quão grande é a necessidade de

ser implementado um Sistema de Ouvidoria, pois percebe que é grande o número de reclamações nas redes sociais e na TV a respeito da cidade. Como a cidade de Assis não tem estrutura para atender todas as manifestações realizadas pelos servidores e pelos cidadãos, através de cartas, e-mails, a aplicação do projeto será muito útil não apenas para a própria Prefeitura, mas também para todas as pessoas, pois através do sistema essas pessoas serão a voz da população dentro da Prefeitura.

Criar um Sistema de Ouvidoria que possibilite todos os cidadãos consultar o andamento de suas manifestações, podendo também enviar reclamações, sugestões, e até mesmo elogios, tendo a opção de fazer uma manifestação anônima, onde não serão exigidos dados cadastrais, porém não terão o privilégio de acompanhar todo o processo de forma detalhada.

1.4 – PÚBLICO ALVO

O sistema desenvolvido será direcionado aos cidadãos, proporcionando-lhes um ambiente onde poderão fazer o acesso para registrar ou consultar suas manifestações. O sistema será inicialmente utilizado por um órgão público, mas também com algumas modificações, terá a capacidade de ser utilizado tanto num ambiente interno de uma empresa (intranet) ou até mesmo pode, a fim de registrar e fornecer informações a respeito de reclamações, serviços, elogios, onde essas informações serão enviadas para um ouvidor, e logo depois de encaminhada para o setor responsável para a resolução do problema, retornando então para o cidadão ou cliente, em forma de uma manifestação.

1.5 MOTIVAÇÃO DO PROJETO

O intuito deste trabalho é propor para toda a população de Assis e região, ou até mesmo com algumas pequenas adaptações poderá ser utilizado em uma empresa privada que queira adotar um Sistema de Ouvidoria informatizado para

gerenciamento de manifestações, independente de onde estejam explorar os recursos que a internet oferece, com total segurança e qualidade, promovendo satisfação entre ambas as partes.

Além de ser uma maneira cômoda para as pessoas exporem suas ideias e fazer suas reclamações, acaba sendo uma forma organizada de ter as informações registradas, pois o objetivo do Sistema de Ouvidoria é procurar ser desenvolvido baseado no código de defesa do consumidor, passando maior segurança para os usuários.

Como já mencionado, o que se espera é um sistema de grande eficiência e segurança, onde seu custo de desenvolvimento acaba sendo viável, pois grande parte das ferramentas utilizadas para o seu desenvolvimento também são utilizadas por grandes empresas, demonstrando então seu grande potencial. A linguagem de programação utilizada será C# (C Sharp), pois será desenvolvido em ambiente web, oferecendo flexibilidade dentre os vários tipos de sistemas operacionais existentes no mercado, proporcionando grande segurança e consistência ao programador. O Astah será utilizado para auxiliar na modelagem do sistema, pois é uma ferramenta visual que possibilita a visualização dos diagramas.

CAPÍTULO 2 – CONCEITOS TEÓRICOS

2.1 - OUVIDORIA

Ao longo dos anos, tem crescido o número de serviços prestados por organizações de médio e grande porte, e conseqüentemente o número de informações também vem crescendo na mesma proporção, tendo então a necessidade de se manter esses dados organizados para que não haja dificuldade na identificação dos possíveis problemas decorrentes dessa enorme expansão da tecnologia e serviços oferecidos. Como também é necessária a comunicação entre as pessoas e as organizações, tem crescido o número de ferramentas que auxiliam essa tarefa, mas nem todas essas ferramentas são consideradas como um processo de comunicação bilateral, ou seja, exige que a transmissão e recepção de ambas as partes para cumprir seu papel.

A Ouvidoria se tornou uma ferramenta para auxiliar as organizações e empresas a se aproximarem de seus clientes, de modo que as manifestações decorrentes das necessidades dos usuários venham provocar uma melhoria dos serviços oferecidos por essas organizações, permitindo que a população participe da gestão pública, estreitando a relação entre a sociedade e o Estado quando se tratando de órgãos públicos.

Na Ouvidoria, todas as informações recebidas através das manifestações realizadas devem servir de como fonte de dados, para que o órgão a qual fosse direcionado use de base para detectar a existência de possíveis problemas, e conseqüentemente a melhoria na qualidade induzidas pelas mudanças que poderão ser realizadas.

2.2 – DEFININDO SISTEMA

A teoria geral dos sistemas foi elaborada pelo biólogo alemão Ludwig Von Bertalanffy, no início da década de 1930, porém apenas foi publicada em 1968. E foi a partir de suas teorias que diversos outros autores começaram a desenvolver a respeito do tema, e hoje Teoria dos Sistemas se encontra difundida em diversas áreas da ciência, se expandindo a diversas disciplinas usadas na atualidade.

De acordo com o dicionário Aurélio, a palavra sistema tem o significado de reunião de princípios coordenados de modo a formar uma doutrina, ou seja, é um conjunto de elementos conectados entre si, de forma organizada, com um objetivo geral a ser atingido. Bertalanffy (1975, p. 52) explica como essa teoria é aplicada em diversas disciplinas:

A ciência moderna é caracterizada por sua crescente especialização, determinada pela enorme soma de dados, pela complexidade das técnicas e das estruturas teóricas de cada campo. Assim, a ciência está dividida em inúmeras disciplinas que geram continuamente novas subdisciplinas. Em consequência, o físico, o biólogo, o psicólogo, e o cientista social estão, por assim dizer, encapsulados em seus universos privados, sendo difícil conseguir que uma palavra passe de um casulo para outro.

Para definir um Sistema Computacional, é necessário que se encontre os atores, fazendo as seguintes perguntas:

- Quem será o usuário do sistema?
- Quem irá instalar o sistema?
- Quem será responsável pelo suporte do sistema durante o uso?
- Quem irá fazer a manutenção dos dados do sistema?
- Quem irá desligar o sistema?

Quando se trata de um Sistema Computacional, há vários fatores relevantes que se devem colocar em questão, pois é muito grande a quantidade de componentes necessários para o processamento das informações, como por exemplo, os hardwares e outros componentes importantes para a execução da aplicação, pois são neles que serão armazenados todo o conteúdo do sistema.

2.3 - INFORMAÇÃO

Os dicionários definem informação como o ato de informar, onde por esta visão, a informação parece ser vista como algo que advém de uma ação, ou seja, a informação acaba sendo o resultado constituído de uma mensagem, originada de um fenômeno ou evento, capaz de resolver problemas e auxiliar na tomada de decisões.

A palavra informação, e segundo o dicionário Michaelis (<http://michaelis.uol.com.br>) possui como definição o *ato ou efeito de informar, de transmissão de notícias, comunicação, transmissão de conhecimento etc.*

Com a evolução da informática, que por sinal ocorre de forma constante, faz com que percebamos os principais fatores decorrentes da “Era da Informação” através dos seguintes processos:

- O ritmo com que as máquinas e as informações cresceram até surgimento da internet;
- O impacto que a evolução tem causado na sociedade, abrangendo todos os aspectos, como políticos, econômicos, sociais, culturais, etc;
- O surgimento de novas tecnologias para armazenamento de informações, métodos para possibilitar a comunicação entre eles e a forma de transformar essas informações em dados.

Percebe-se o quanto é importante que as informações necessitem ser tratadas, viabilizando a implementação de novos métodos e sistemas exclusivos, otimizando então os processos e diminuindo os custos. A seguir, a visão de Silva (2001, p.03) sobre essa ideia:

Nos últimos tempos, as empresas de uma forma geral passaram a se preocupar com a racionalização de seu sistema de trabalho, buscando alternativas para enfrentar o acirramento das concorrências. A filosofia empresarial é produzir mais a um menor custo.

Observando sob esse prisma, a área de processamento de dados jamais poderia ficar atrás e, portanto, a otimização dos processos de geração de informações passou a ser analisada de uma forma mais séria e profissional.

2.4 – SISTEMAS DE INFORMAÇÃO

Nos dias de hoje é muito comum ouvir falar sobre a Era da Informação, também conhecida como Era Digital, que baseia no período após a Era Industrial, onde começa a acontecer o grande crescimento das informações e o surgimento de novas tecnologias, onde a forma dessa nova era passa a se parecer cada vez menos com a forma mecanizada adotada na Era Industrial.

Com esse grande crescimento de informações, passou-se a dar mais valor não apenas na capacidade física do homem, mas sim num homem como um todo. Com isso cresce a necessidade de métodos para o gerenciamento dessas informações, ou seja, sistemas capazes de coletar, processar, armazenar e fornecer resultados de forma automática e com grande eficiência.

Como o próprio nome descreve, Sistema de Informação é o termo usado para descrever um sistema, que por sua vez, manual ou automático, com a interação de homens ou máquinas, torna-se um mecanismo capaz de coletar, processar, armazenar e disseminar informações específicas para um determinado usuário.

O Desenvolvimento de Sistema de Informação (DSI) é um processo de mudança, que tem o objetivo de melhorar um subsistema, e por consequência promover melhoria do Sistema da Informação.

Segundo Rocha (2008, p.03), num processo de DSI deve ser considerado três domínios de mudança: tecnologia, linguagem e organização. Sobre esses domínios, Rocha (2008, p.04) diz o seguinte:

A tecnologia cobre os meios físicos e o saber fazer técnico pelo qual as tarefas de processamento de dados são consumadas. Por linguagem é entendida qualquer forma de representação simbólica que transporta significado. Isto inclui conversação, mas também sistemas de processamento de dados convencionais. Finalmente, organização, cobre elementos tais como procedimentos e arranjos de trabalho, papéis, posições, poder, valores, normas e cultura bem como competências técnicas e aptidões para realizar processos de trabalho.

A utilização de Sistemas de Informação tem se tornado cada vez mais necessária, porque há necessidade na agilidade de tomadas de decisões nas organizações, pois como os Sistemas tem evoluído muito ao longo do tempo, essa tarefa vem sendo cumprida de forma fácil e eficiente por eles.

Com os Sistemas Informatizados, as informações passam a ficar cada vez mais organizadas, facilitando o uso e a recuperação de informações independentes da distância em que se encontram, tamanho e formatos que as mesmas se encontram.

A figura a seguir demonstra um simples esquema de Sistema, onde há entrada de dados, subsistemas, processamento e saída das informações com seu respectivo feedback:

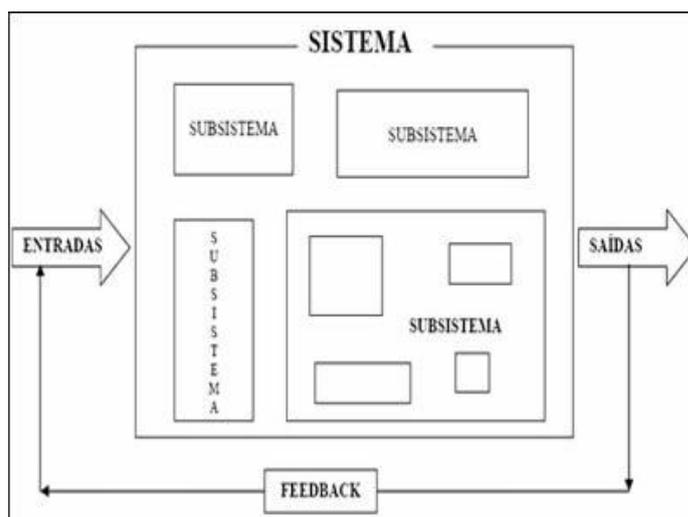


Figura 1 – Exemplo do fluxo de um sistema.(Fonte: <http://www.brasilecola.com/upload/e/sistema.jpg>)

2.5 – ANÁLISE DE SISTEMAS

A Análise de Sistemas é a atividade inicial do processo de desenvolvimento de sistemas, com a finalidade de estudar todos os processos existentes para encontrar o melhor caminho, de forma lógica, para que a informação possa ser processada, utilizando de soluções padronizadas de forma que o computador tenha a capacidade de executar.

Segundo Rocha (2008, p.10), *o termo é por vez substituído pelo termo engenharia de requisitos, onde a adoção do novo nome deve-se à preocupação de se pretender incorporar uma orientação de engenharia, adicionando-lhes mais vigor e disciplina.*

O objetivo principal da Análise de Sistemas é levantar os requisitos referentes aos dados usados durante o desenvolvimento do projeto do sistema, deixando bem claro o que deve ser feito, ou seja, nessa etapa o usuário apresenta ao Analista todas as suas necessidades.

Aproveitando as palavras de (Rocha, 2008, p.10) *O uso do termo engenharia implica que técnicas sistemáticas e repetíveis devam ser usadas para assegurar que os requisitos dos sistemas são relevantes, consistentes e completos.*

A partir dos dados coletados, o Analista estuda os mesmos e procura resolver os possíveis problemas que poderão ocorrer durante o desenvolvimento do projeto, tendo como resultado final da sua análise uma sequência lógica na qual servirá como base para o projeto.

No desenvolvimento do projeto, é importante que o analista não tenha ficado com dúvidas sobre os requisitos levantados, para que possa apresentar ao projeto exatamente o que, e como deve ser feito, pensando nos custos e seus benefícios que terão durante o desenvolvimento, escolhendo então os tipos de softwares que poderão ser usados. Silva (2001, p. 34) afirma sobre isso:

A escolha da melhor alternativa é bastante pessoal e deve obedecer a uma rigorosa análise técnica e econômica e além do mais, depende de disponibilidade financeira e da visão empresarial de cada empreendedor. As propostas de soluções são elaboradas em forma de relatório e cada uma delas deverá discriminar estimativa de custo e benefícios.

Quando o nível de comunicação entre o analista e o usuário é muito baixo, os requisitos podem não ser compreendidos corretamente e conseqüentemente o analista não poderá identificar os problemas subsequentes. Essa é uma das várias barreiras que há entre o analista e os usuários do sistema. Aproveitando as palavras de Silva(2001, p.36) , vemos qual é a função do analista nessa etapa do projeto:

Para conclusão do projeto detalhado, o analista deverá fazer uma breve exposição ao usuário de como será o funcionamento de todo o sistema, permitindo-lhe a visualização de suas principais funções e definindo como deverá ser o treinamento dos usuários do novo sistema.

2.6 – ENGENHARIA DE SOFTWARE

O desenvolvimento de softwares tem sido uma atividade que vem crescendo de forma constante, e com grande importância na sociedade, pois a utilização da informática nas diversas áreas vem fazendo com que haja uma maior demanda para a produção de soluções informatizadas.

Quando se fala em desenvolvimento de softwares, inicialmente parece ser uma tarefa difícil para as pessoas que estão começando no ramo, contudo, acabam confundindo programação com desenvolvimento. Isso acontece porque no início dos estudos as pessoas acabam desenvolvendo habilidades que exercitam seu raciocínio lógico através do estudo de algoritmos e estruturas de dados.

Portanto, vai chegar a um ponto em que o programador não irá conseguir mais resolver determinados problemas, devido ao tamanho de sua complexidade. Comparando com outras áreas da engenharia, com, por exemplo, a Engenharia

Civil, que para construir um prédio é necessário a elaboração de um projeto antes, pois há a necessidades de estudos aprofundados que um pedreiro não é capaz de fazer para que a construção seja bem feita.

Dai então surge a Engenharia de Software com a finalidade de melhorar a qualidade dos softwares, e trazer soluções para aumentar a produtividade no desenvolvimento, através de técnicas, métodos, estabelecendo processos com ferramentas específicas para tal finalidade. Pressman (2011, p.39) fala sobre detalhes indispensáveis a respeito do conceito de Engenharia de Software:

A engenharia de software é uma tecnologia em camadas. (...) qualquer abordagem de engenharia (inclusive engenharia de software) deve estar fundamentada em um comprometimento organizacional com a qualidade. A gestão da qualidade total Seis Sigma e filosofias similares promovem uma cultura de aperfeiçoamento contínuo de processos, e é esta cultura que, no final das contas, leva ao desenvolvimento de abordagens cada vez mais efetivas na engenharia de software. A pedra fundamental que sustenta a engenharia de software é o foco na qualidade.

Já que o foco da Engenharia de Software é melhorar a qualidade do software, deve então analisar o que é qualidade de software.

Quando se trata de qualidade de software, há dois pontos a serem observados: o ponto de vista do usuário do sistema e do desenvolvedor, pois para o usuário, um software de qualidade é aquele com boa interface, fácil de usar, com grande eficiência e de confiabilidade. Já para o desenvolvedor tem que ser um projeto bem documentado, ou seja, fácil de manter, possibilitando também um fácil entendimento para o desenvolvimento atendendo os requisitos do usuário. Dai então se dá a importância da Engenharia de Software e de um bom levantamento de requisitos.

2.7 – UML

A UML (*Unified Modeling Language*), onde traduzindo seria Modelagem Unificada de Modelagem, e como o próprio nome diz, é uma linguagem de modelagem visual, para especificação, construção e documentação dos artefatos que contidos em um *software*, incluindo sua estrutura e *design*.

Guedes (2011, p.19) expõe a UML com a seguinte definição:

(...) é uma linguagem visual utilizada para modelar softwares baseados no paradigma de orientação a objetos. É uma linguagem de modelagem de propósito geral que pode ser aplicada a todos os domínios de aplicação. Essa linguagem tornou-se nos últimos anos, a linguagem-padrão de modelagem adotada internacionalmente pela indústria de engenharia de software.

Com um grande número de ferramentas existentes no mercado, a UML ajuda muito os engenheiros de *software* no momento de levantar os requisitos da futura aplicação e projetar em formas de diagramas as soluções para o desenvolvimento do mesmo.

Lembrando que a UML não é uma linguagem de programação, mas sim uma linguagem de modelagem e visual, baseado no conceito de orientação a objetos. Aproveitando as palavras de Guedes(2011, p19), vimos a seguinte abordagem a respeito disso:

(...) UML não é uma linguagem de programação, e sim uma linguagem de modelagem, uma notação, cujo objetivo é auxiliar os engenheiros de software definirem as características do sistema, tais como seus requisitos, seu comportamento, sua estrutura lógica, a dinâmica de seus processos e até mesmo suas necessidades físicas em relação ao equipamento sobre qual o sistema deverá ser implantado.

A análise dos requisitos irá coletar todas as necessidades dos usuários do sistema, através das funções que se encontram nos casos de uso, e é nessa fase que se deve encontrar a definição correta do software ou dos próprios requisitos do sistema.

O *design* passa a ser então a visualização do resultado obtido pela análise, ou seja, serão detalhados todos os possíveis problemas que devem ser eliminados para que então possa entrar na fase de programação. Dai então os problemas na fase de desenvolvimento deverão ser descritos conforme a linguagem de programação escolhida. Por isso é importante a modelagem antes de começar a desenvolver o software, pois o usuário terá a oportunidade de conhecer seu sistema mesmo antes de ser desenvolvido, eliminando então as chances de ter que alterar as funcionalidades do sistema depois de pronto. Guedes (2011, p.20) ressalta a importância da modelagem antes da fase de desenvolvimento:

Na realidade, por mais simples que seja, todo e qualquer sistema deve ser modelado antes de se iniciar sua implementação, entre outras coisas, porque os sistemas de informação frequentemente costumam ter a propriedade de “crescer”, isto é, aumentar em tamanho, complexidade e abrangência. Muitos profissionais costumam afirmar que sistemas de informação são “vivo”, porque nunca estão completamente finalizados. Na verdade, o termo correto seria “dinâmicos”, pois normalmente os sistemas de informação estão em constante mudança.

Dentre as várias divisões contidas na UML, podemos destacar os diagramas, que por sinal são os componentes mais importantes para a visualização do sistema visualmente. Os diagramas mais importantes são: Diagrama de Caso de uso, Diagrama de Atividades, Diagrama de Classes e o Diagrama de Sequência.

2.7.1 – Diagrama de Caso de Uso

São os diagramas responsáveis por determinar as especificações funcionais de um sistema, ou seja, auxiliam a comunicação entre os desenvolvedores do sistema e seus clientes, através da compreensão dos cenários que dita qual a funcionalidade do sistema no ponto de vista do usuário. Estes diagramas apresentam em sua estrutura os Atores, os Casos de Uso e os Relacionamentos que ocorrem entre eles. Guedes (2011, p.30) descreve o diagrama de caso de uso do seguinte ponto de vista:

(...) é o diagrama mais geral e informal da UML, utilizado normalmente nas fases de levantamento e análise de requisitos do sistema, embora venha a ser consultado durante todo o processo de modelagem e possa servir de base para outros diagramas. Apresenta uma linguagem simples e de fácil compreensão para que os usuários possam ter uma ideia geral de como o sistema irá se comportar. Procura identificar os atores (usuários, outros sistemas ou até mesmo algum hardware especial) que utilizarão de alguma forma o software, bem como os serviços, ou seja, as funcionalidades que o sistema disponibilizará aos atores, conhecidas nesse diagrama como casos de uso.

2.7.2 – Diagrama de Atividades

O Diagrama de Atividades é o diagrama que representa o fluxo das atividades conduzindo em um único processo, mostrando como uma atividade depende da outra para ser executada. Esses processos podem ser executados por computadores, pessoas ou até mesmo outros componentes do sistema.

O diagrama pode ser usado para descrever diversos tipos de processos. Vejamos alguns exemplos desses processos:

- Um algoritmo de um sistema;
- Sequências que permitem interações entre componentes;
- Etapas que podem ser executadas em casos de uso;

- Fluxo de um trabalho, ou um processo de negócios entre o usuário e o seu sistema.

Esse tipo de diagrama acaba não sendo apenas importante na modelagem do sistema, mas também pode ser usado na construção de softwares executáveis por meio da engenharia reversa de produção.

2.7.3 – Diagrama de Classes

O Diagrama de Classes é definido como um conjunto de objetos que representa a estrutura e a relação das classes, dando como base para o desenvolvimento dos objetos em si. Entretanto, sua estrutura é composta por atributos, métodos e as reações entre eles, onde os atributos definem as características da classe, e os métodos representam as ações que os objetos podem realizar. A seguir podemos ver a definição dada por Guedes (2011, p.31):

O diagrama de classes é provavelmente o mais utilizado e é um dos mais importantes da UML. Serve de apoio para a maioria dos demais diagramas. Como o próprio nome diz, define a estrutura das classes utilizadas pelo sistema, determinando os atributos e métodos que cada classe tem, além de estabelecer como as classes se relacionam e trocam informações entre si.

Pelo fato de apontar diretamente as classes, o diagrama de classe acaba sendo considerado como o principal dentre os diagramas, pois apontam também as interfaces e os relacionamentos entre as mesmas, com a capacidade de ilustrar a funcionalidade e as características determinadas para cada classe, mostrando claramente o relacionamento entre elas.

2.7.4 – Diagrama de Sequência

Esse diagrama é usado na UML para representar as sequências de processos, mostrando as mensagens trocadas entre os objetos, descrevendo a maneira de como o comportamento de cada objeto tem a capacidade de colaborar com o comportamento de outros dentro de um determinado contexto. De uma maneira mais clara e detalhada, Guedes (2011, p. 31) descreve o que é Diagrama de Sequência:

(...) um diagrama comportamental que preocupa-se com a ordem temporal em que as mensagens são trocadas entre os objetos envolvidos em um determinado processo. Em geral, baseia-se em um caso de uso definido pelo diagrama de mesmo nome e apoia-se no diagrama de classes para determinar os objetos das classes envolvidas em um processo. Um diagrama de sequência costuma identificar o evento gerador do processo modelado, bem como o ator responsável por esse evento, e determina como o processo deve se desenrolar e ser concluído por meio da chamada de métodos disparados por mensagens enviadas entre os objetos.

É válido lembrar que existem diversos outros diagramas usados na UML, mas que não é necessariamente obrigatório usar todos para modelar um sistema, pois o que define a quantidade e tipos de diagramas usados será a complexidade do sistema, de acordo com suas características e suas funções.

2.8 – ORIENTAÇÃO A OBJETOS

Desde que foi inventado o computador, a programação tem sofrido drásticas mudanças, pois a necessidade de diminuir a complexidade dos programas se tornou cada vez maior, fazendo com que cresça a produtividade e o desempenho dos desenvolvedores. Por exemplo, os primeiros computadores eram programados através de chaveamento e utilizando instruções binárias. Logo depois surgiram as

linguagens de alto nível, tornando os programas mais fáceis de entender, sem ter que fazer grandes mudanças no estilo de desenvolver.

Na década de sessenta surge então a programação estruturada, onde pela primeira vez, é possível o desenvolvimento de programas considerados complexos, de uma maneira razoavelmente fácil e clara. No início as linguagens mais conhecidas eram o C e Pascal.

Cada passo que a programação dava, eram melhorados os elementos e métodos utilizados anteriormente, permitindo novas combinações tornando-os cada vez mais avançados. Como os novos programas ficaram muito complexos, o método estruturado já não estava mais resolvendo os problemas, conseqüentemente não funcionava bem. Para resolver os problemas desse antigo método, foi criada então a orientação a objetos, aproveitando então as melhores ideias que eram usadas na programação estruturada, combinando com novos conceitos de desenvolvimento.

A análise, projeto e programação orientada a objetos tiveram seu início na década de setenta, onde diferentemente da análise e projetos estruturados, na orientação a objetos os processos que são informatizados passam a ser vistos como um conjunto de objetos que interagem entre si para a realização de determinadas funções.

Deste modo, Velloso (2004, p.76) define orientação a objetos da seguinte maneira:

Programação orientada a objetos: (OOP – Object Oriented Programming) é a programação através da qual os programas recorrem a áreas ou partições de memória onde encontram objetos (dados + procedimentos) representativos de cada realidade.

Quando é falado de orientação a objetos e do método de desenvolvimento estruturado, atualmente a orientação a objetos é mais forte, por isso se dá a importância de diferenciar os dois métodos. No entanto, é importante ressaltar que orientação a objetos terá um melhor aproveitamento e rendimento de acordo com a

mentalidade do desenvolvedor, pois dependendo da maneira de programar, podem-se conseguir programas orientados a objeto com características de estruturados e vice-versa.

A seguir, será descrito de forma clara e sucinta alguns dos principais elementos usados na programação orientada a objetos.

2.8.1 – Classe

Classe é o termo utilizado em linguagens orientadas a objetos que possuem as características (atributos), ou seja, é composto de um conjunto de dados de forma estruturada, caracterizados por suas comuns propriedades.

As classes de objetos são descritas como um conjunto de objetos através de um conjunto de atributos e métodos. Esse conjunto de objetos então podem compartilhar as mesmas operações. Boratti (2001, p.18) diz:

Uma classe define as características de um grupo de objetos, isto é, define como serão as instâncias pertencentes a ela. Assim, uma classe especifica quais serão os atributos dos objetos que pertencerem a mesma, bem como, que serviços qualquer objeto da classe poderá executar.

2.8.2 – Atributo

Um atributo é caracterizado pelo valor que o objeto apresenta, ou seja, são os elementos que definem a estrutura da classe, basicamente as características do objeto. Boratti (2002, p. 14) define atributos como “*as características de composição de um objeto (que) estão associadas a sua constituição e definem a sua estrutura.*”.

2.8.3 – Método

Os métodos dentro de uma classe representam as operações que os objetos pertencentes a tal classe é capaz de executar, podendo ser comparado a um procedimento, ou propriamente dito, uma função das linguagens de programação. Complementando o conceito, Boratti (2002, p.19) descreve o método como um *“conjunto de ações que devem ser executadas pelo computador, representando a execução do serviço pelo objeto”*.

Os métodos são rotinas, que quando executadas, deve realizar alguma tarefa, como por exemplo, alterar o conteúdo da variável de um objeto, fazendo com que esses objetos se comuniquem entre si através de mensagens, e quando necessário, passando parâmetros ao método permitindo especificar alguma função a ser executada pelo mesmo. Lembrando que as variáveis do objeto só podem ser alteradas por métodos definidos dentro da sua própria classe.

2.8.4 – Objeto

Na orientação a objetos, um objeto é uma abstração de software onde pode representar coisas que sejam reais ou virtuais. De um modo geral podemos interpretar os objetos como sendo os físicos do nosso mundo real, tal como: cachorro, carro, avião, computador, casa, prédio, etc., sendo dessa forma mais fácil a compreensão do conceito na programação.

Dando como exemplo de um objeto uma moto, onde possuem suas propriedades, como cilindradas, limite de passageiros, velocidade. Tal objeto também apresenta seus procedimentos, como acelerar, parar, ligar e desligar.

Os objetos usam mensagens como meio de comunicação, e quando esse objeto deseja alguma outra tarefa contida em outro objeto, ele então envia a mensagem contendo o objeto de origem, o objeto de destino, o nome do método existente no objeto de destino, e quando necessário algum parâmetro para especificar a função a ser executada pelo método.

2.9 – BANCO DE DADOS

Com o aumento do volume de dados que são processados e transmitidos todos os dias, fez com que aumentasse a necessidade de se possuir uma maneira segura e eficaz de se armazenar essas informações, fazendo com que seja necessário adquirir novos conhecimentos, possibilitando a melhora nas tomadas de decisões, consideradas como estratégias que viabilizam a utilização do banco de dados. Para obter essas estratégias para que haja um melhor desempenho na organização, é necessário um banco de dados consistente.

Resumidamente, banco de dados se consiste num conjunto de informações, que de forma organizada e relacionadas, dão um sentido para sua existência, possibilitando o gerenciamento dos dados para atender a necessidades das empresas e organizações.

Teorey (2007, p.02) define banco de dados da seguinte forma:

Um banco de dados é um objeto mais complexo; é uma coleção de dados armazenados e inter-relacionados, que atende às necessidades de vários usuários dentro de uma ou mais organizações, ou seja, coleções inter-relacionadas de muitos tipos diferentes de tabelas. As motivações para usar banco de dados em vez de arquivos incluem a maior disponibilidade a um conjunto diversificado de usuários, a integração de dados para facilitar o acesso e a atualização de transações complexas, e a menor redundância de dados

Uma das principais razões para que se utilize a metodologia de banco de dados, é ter o controle total dos dados de uma organização, onde de forma segura possa ser feito através de programas específicos para tal finalidade.

Referente à sua estrutura, Velloso (2004, p 89) expõe que o banco de dados apresenta as seguintes formas, conhecidas como modelo de banco de dados:

- Modelo em rede;
- Modelo relacional;
- Modelo hierárquico;

A respeito desses modelos, Velloso (2004, p.89) resume os três conceitos da seguinte maneira:

No modelo hierárquico, a informação é organizada em níveis. Um registro em nível superior pode ter um, mais de um, ou não ter “filhos”, no nível imediatamente inferior. Para cima, é diferente: qualquer registro pode ter apenas um “pai”.

O acesso é feito, pelos ramos, de nó em nó, nos dois sentidos.

No modelo em rede, cada registro mestre é acessado por uma chave. Dele, vai-se sequencial ou diretamente aos registros membros.

No modelo relacional estabelece-se a ligação entre listas ou relações, por meio de ponteiros.

Antigamente, utilizavam-se sistemas de arquivos de banco de dados do próprio sistema operacional como forma de armazenamento dessas informações. Atualmente é utilizado o Sistema Gerenciador de Banco de Dados (SGBD) para o gerenciamento dessas informações.

Os Sistemas Gerenciadores de Banco de Dados, como o próprio nome diz, são conjuntos de softwares responsáveis pelo gerenciamento do banco de dados, onde através de uma interface gráfica ou via código, possibilite que seus usuários tenham a capacidade de alterar, incluir, excluir, consultar, ou seja, fazer qualquer tipo de alteração nos dados armazenados na base de dados. Complementando tal conceito, Teorey (2007, p.02) descreve o SGBD da seguinte forma:

Um sistema gerenciador de banco de dados (SGBD) é um sistema de software genérico para manipular bancos de dados. Um SGBD admite uma visão lógica (esquema e sub-esquema);visão física (métodos de acesso,

clustering de dados); linguagem de definição de dados; linguagem de manipulação de dados; e utilitários importantes, como gerenciamento de transação e controle de concorrência, integridade de dados, recuperação de falhas e segurança.

Com a necessidade de viabilizar a implementação no modelo de dados relacional, foi criado então na década de setenta a linguagem Sequel, com o objetivo de padronizar uma linguagem para ser usada nesse tipo de modelo. Logo depois de alguns ajustes e aprimoramentos, surgiu a SQL (Structured Query Language), ou seja, uma linguagem estruturada de consulta, tornando-se então linguagem universal e padrão para banco de dados. Através das palavras de Velloso (2004, p.92), vemos a importância da tal linguagem:

A linguagem de manipulação de dados que vem se impondo, e em particular, dominando o segmento de microcomputação é a Structured Query Language ou Linguagem Estruturada de Consulta. Com a implantação das redes do tipo cliente servidor e dos bancos de dados corporativos no modelo relacional, tem sido ela utilizada preferencialmente para que cada estação tenha acesso às informações de seu interesse. Independentemente do SGBD e do equipamento que o ambiente utilize.

Hoje a linguagem SQL não é reconhecida conceitualmente como uma linguagem em si, mas um conjunto de facilidades, como recursos de acesso, manutenção de dados e definições.

2.10 – LINGUAGEM DE PROGRAMAÇÃO

Desde o surgimento dos primeiros computadores percebe-se a necessidade de encontrar uma forma padronizada, aonde através de instruções, venha auxiliar a comunicação do homem com a máquina.

Um acervo de palavras com instruções lógicas, um conjunto de regras sintáticas e semânticas de forma que possa ser facilmente interpretada pela máquina definem a

estrutura de um programa. Dessa forma surge a linguagem de programação, que passa a ser um conjunto de palavras, composta por regras e instruções, que possibilite que um programador indique a uma máquina quais os dados a máquina deve utilizar, como e onde serão transmitidos e armazenados, ou quaisquer outras ações dependendo da circunstância em que se encontram.

No início do surgimento da programação, foi encontrada muita dificuldade em questão de desenvolvimento e armazenamento, pois as máquinas não forneciam condições físicas para que fossem desenvolvidos programas complexos e eficientes, devido a baixa capacidade de armazenamento e processamento dos dados.

Atualmente podemos encontrar milhares de linguagens de programação no mercado, cada uma com sua funcionalidade e particularidade, embora seja possível desenvolver programas com a mesma funcionalidade, porém utilizando linguagens diferentes. A seguir, será apresentada uma figura com uma lista contendo as linguagens mais utilizadas no ano de 2012, encontrada no site <http://info.abril.com.br>:

Posição	Linguagem de programação	Avaliação
1	Java	17.050%
2	C	16.523%
3	C#	8.653%
4	C++	7.853%
5	Objective-C	7.062%
6	PHP	5.641%
7	(Visual) Basic	4.315%
8	Python	3.148%
9	Perl	2.931%
10	JavaScript	2.465%
11	Delphi/Object Pascal	1.964%
12	Ruby	1.558%
13	Lisp	0.905%
14	Transact-SQL	0.846%
15	Pascal	0.813%
16	Visual Basic .NET	0.796%
17	PL/SQL	0.792%
18	Logo	0.677%
19	Ada	0.632%
20	R	0.623%

Figura 2 – Linguagens de programação mais usadas em 2012.
(Fonte: <http://info.abril.com.br/noticias/ti/veja-as-20-linguagens-de-programacao-mais-populares-17022012-26.shl> >> Acesso em 16/06/2013)

CAPÍTULO 3 - MÉTODOS DE DESENVOLVIMENTO E FERRAMENTAS

As ferramentas utilizadas para o desenvolvimento do Sistema de Ouvidoria são: Visual Studio 2012, Astah, SQL Server 2008, SQL Management Studio 2008 e o servidor IIS. Será abordada neste capítulo uma breve descrição de cada ferramenta utilizada, de forma objetiva e de fácil entendimento. Como linguagem de programação, será usado o C# (pronuncia-se C Sharp).

3.1 – C# (C SHARP)

Criada por Anders Hejlsberg entre os anos de 1999 e 2000, a linguagem C# (pronuncia-se C Sharp) é uma linguagem de programação orientada a objetos, criada pela Microsoft, baseada na linguagem C++ e Java, que veio com o intuito de oferecer uma linguagem robusta e simples, aproveitando os melhores recursos de cada linguagem, ainda com a capacidade de ser utilizada para desenvolvimento de aplicativos para diversos tipos de dispositivos.

Sobre essa nova linguagem de programação, Battisti (2006, p.74) diz o seguinte:

“Para resumir em poucas palavras quais os objetivos da Microsoft com a criação do C#, eu diria o seguinte: Uma nova linguagem, criada para ter simplicidade e facilidade de desenvolvimento do Visual Basic, aliadas com o poder do C++.”

3.2 – FRAMEWORK .NET

O Framework .NET é nada mais do que um ambiente de execução com a função de gerenciar alguns aplicativos, proporcionando vários tipos de serviços ao aplicativos que se encontram em tempo de execução.

Dentre seus principais componentes, encontra-se um mecanismo capaz de manipular tais aplicativos em execução, e também uma vasta coleção de bibliotecas de classes e códigos reutilizáveis. Battisti (2006, p.16) diz que “*O Framework .NET foi todo projetado já tendo a Internet como objetivo, diferente de outras plataformas que foram adaptadas para a Internet*”

3.3 – VISUAL STUDIO

O Visual Studio é um IDE (Integrated development Environment) desenvolvido pela Microsoft, composta de inúmeras ferramentas com capacidade de escrever códigos em diversas linguagens de programação. Seu grande potencial proporciona a possibilidade de desenvolvimento de projetos de grande complexidade, de forma simples e com resultados de alta qualidade.

Dentre essas diversas ferramentas que compõem o Visual Studio, destaca-se o Framework .NET, onde suas principais linguagens são o C# (C Sharp) e o VB (Visual Basic), possibilitando a criação de aplicações para ambiente web. Para o desenvolvimento do Sistema de Ouvidoria será utilizado o Visual Studio 2012.

3.4 – ASTAH

O Astah é um software utilizado para apoio na modelagem de projetos que utiliza o padrão UML (Unified Modeling Language) para a criação de diagramas, padronizando-os e proporcionando uma melhor visualização das classes e objetos contidos no projeto. Além de criar diagramas UML, o Astah oferece a possibilidade

de exportar seus objetos para diversas linguagens, como por exemplo, para a linguagem Java, minimizando o trabalho do desenvolvedor do sistema.

3.5 – SQL SERVER

O SQL Server é um servidor de banco de dados da Microsoft, que utiliza o modelo relacional para o gerenciamento dos mesmos, disposto de várias funcionalidades que fazem com que ele tenha um diferencial em relação a outros servidores do mercado.

3.6 – SQL MANAGEMENT STUDIO

O SQL Management Studio é um software utilizado para gerenciar e administrar de forma gráfica, todos os componentes do servidor SQL Server, pois suas ferramentas gráficas dão outra visão ao usuário sobre os recursos do próprio servidor, tornando mais fácil a administração do banco de dados, reduzindo a complexidade do ambiente para aplicativos orientados a banco de dados.

Quando se trabalha com o SQL Management Studio junto com o SQL Server, a tarefa de manipular os objetos dentro do banco de dados se torna ainda mais fácil, pois essa é a principal característica desse software de gerenciamento. Neste trabalho será usado o SQL Management Studio 2008.

3.7 – SERVIDOR IIS

Criado pela Microsoft, o IIS (Internet Information Services) é um servidor web desenvolvido para rodar aplicações desenvolvidas pelos próprios sistemas operacionais da Microsoft, como por exemplo, páginas ASP.NET, que é o caso do Sistema de Ouvidoria.

O IIS oferece uma plataforma segura, dando segurança aos sites e serviços nele hospedados, proporcionando aos usuários do sistema web um ambiente confiável.

CAPÍTULO 4 – ANÁLISE DO SISTEMA

4.1 – DADOS SOBRE O CLIENTE

O sistema será desenvolvido para a Prefeitura Municipal de Assis, onde através dos requisitos coletados durante um determinado período, chegou-se a conclusão da necessidade da implementação do Sistema de Ouvidoria.

4.2 – LEVANTAMENTO DE REQUISITOS

Durante o primeiro semestre de 2013, foram feitas várias observações no funcionamento de Sistemas de Ouvidorias em vários tipos de empresas e órgãos públicos, e também analisando a situação da cidade de Assis na atual conjuntura, percebe-se o quanto se necessita um Sistema de Ouvidoria, pois cresce o número de reclamações nas redes sociais e na TV a respeito da cidade.

Durante esse período de observações, foram coletadas informações importantes que irão direcionar todo o desenvolvimento do sistema, ou seja, foram descobertas as necessidades que os cidadãos tiveram em relação ao Sistema de Ouvidoria.

Como o objetivo do sistema é auxiliar os cidadãos em relação à satisfação dos serviços oferecidos pelas organizações, as principais necessidades serão apresentadas a seguir:

- Cadastrar usuários;
- Realizar manifestações.

A partir desses pontos, foi desenvolvido um mapa mental para mostrar de forma visual, uma representação que cada usuário terá no sistema, ajudando a solucionar os possíveis problemas no projeto, e mostrando suas funcionalidades e estruturas.

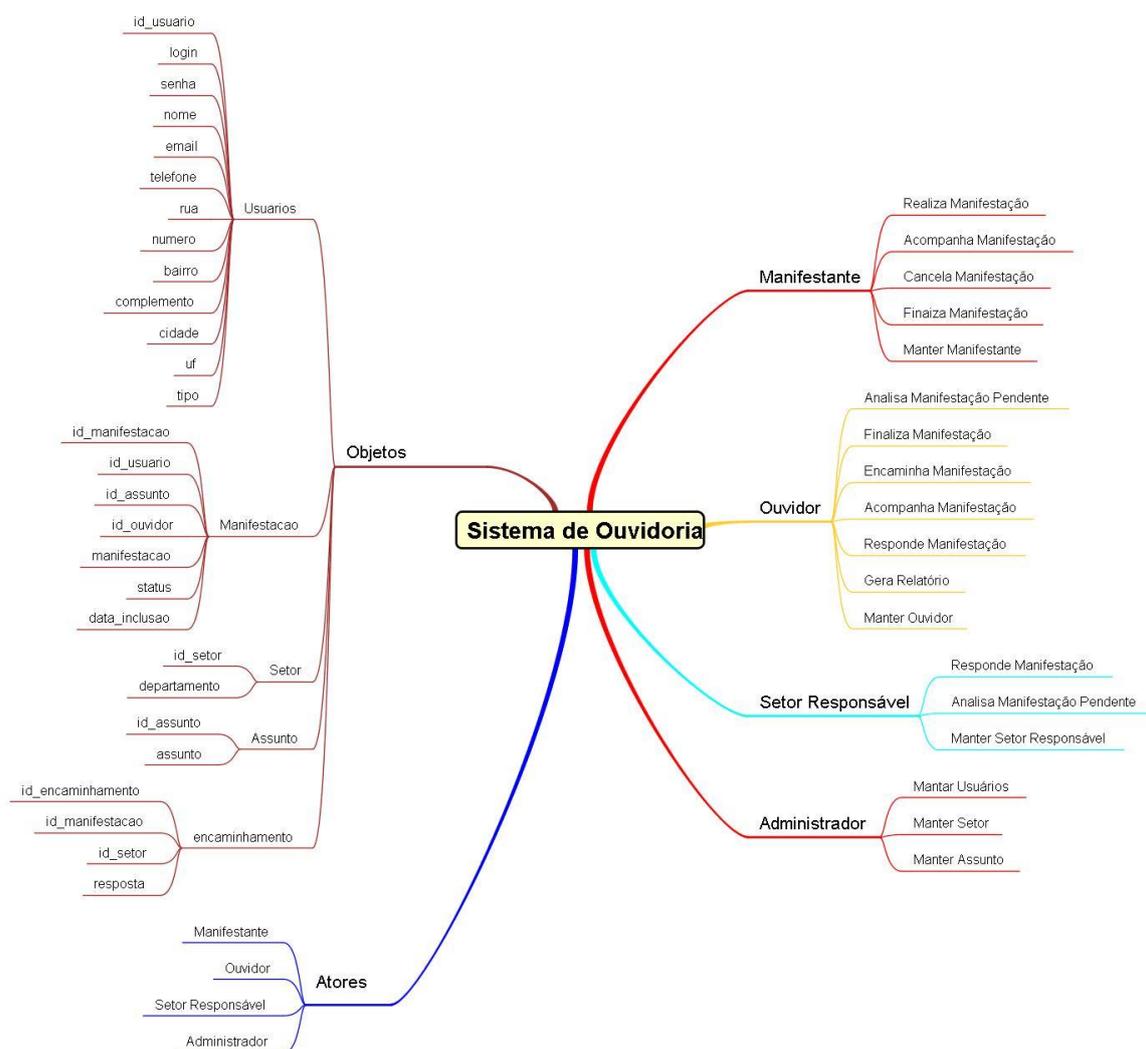


Figura 3 - Mapa mental do Sistema de Ouvidoria.

O sistema foi dividido em três partes: Cadastros, movimentações e relatórios.

Cadastrar / Manter:

- Usuários;
- Manifestações;
- Setores;
- Assuntos.

Movimentar

- Manifestações (acompanhamento, análise, resposta e finalização).

Relatórios

- Manifestações (pendentes, respondidas, não respondidas).

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

Lista de eventos	
Cadastrar / Manter	Usuários Manifestações Setores Assunto
Movimentar	Acompanhamento de manifestações Análise de manifestações Resposta de manifestações Finalização de manifestações
Emitir relatórios	Manifestações pendentes Manifestações respondidas Manifestações não respondidas

Tabela 1 – Lista de eventos

CAPÍTULO 5 – DIAGRAMAS

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

5.1 – DIAGRAMA DE CASOS DE USO – DEFINIÇÃO

Conforme foi dito do capítulo 2, são os diagramas responsáveis por determinar as especificações funcionais de um sistema, ou seja, auxiliam a comunicação entre os desenvolvedores do sistema e seus clientes, através da compreensão dos cenários que dita qual a funcionalidade do sistema no ponto de vista do usuário. Estes diagramas apresentam em sua estrutura os atores, os casos de uso e os relacionamentos que ocorrem entre eles.

O diagrama de casos de uso também auxilia a análise do sistema, onde especifica as funcionalidades do sistema e seu comportamento.

Os atores do diagrama de casos de uso são os elementos externos que interagem de alguma forma com o sistema, onde com um papel específico, estimula os eventos do sistema, iniciando os mesmos.

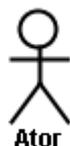


Figura 4 - Representação gráfica de um ator do diagrama de casos de uso.

Para cada caso de uso existente, há um ator que deverá iniciá-lo, onde esse ator deve ser indicado de forma clara para facilitar o entendimento e identificação.

Os atores podem ser representados por seres humanos, hardwares, algum outro processo e até mesmo algum dispositivo.

O relacionamento entre os elementos do diagrama de casos de uso pode ocorrer de diversas formas, podendo ser relacionado entre os próprios atores, entre os atores e casos de uso e também entre casos de uso.

A figura 5 ilustra o tipo de relacionamento que é usado no diagrama de caso de uso do projeto, onde o relacionamento será feito entre atores e casos de uso:

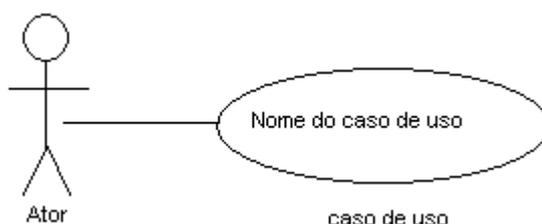


Figura 5 - Representação gráfica do relacionamento entre ator e caso de uso

O relacionamento é representado pela seta que liga o ator ao caso de uso, indicando que é o ator que inicia o caso de uso. O caso de uso é representado pelo balão.

É importante destacar que o caso de uso não irá descrever a forma que o sistema deverá ser escrito, mas sim a forma que ele irá agir depois de pronto.

Sabe-se que um caso de uso tem a finalidade de descrever um fluxo de eventos capazes de realizar uma operação, portanto é necessário que existam pré e pós-condições verdadeiras para a realização dos mesmos.

5.1.1 – Diagrama de casos de uso – Aplicação

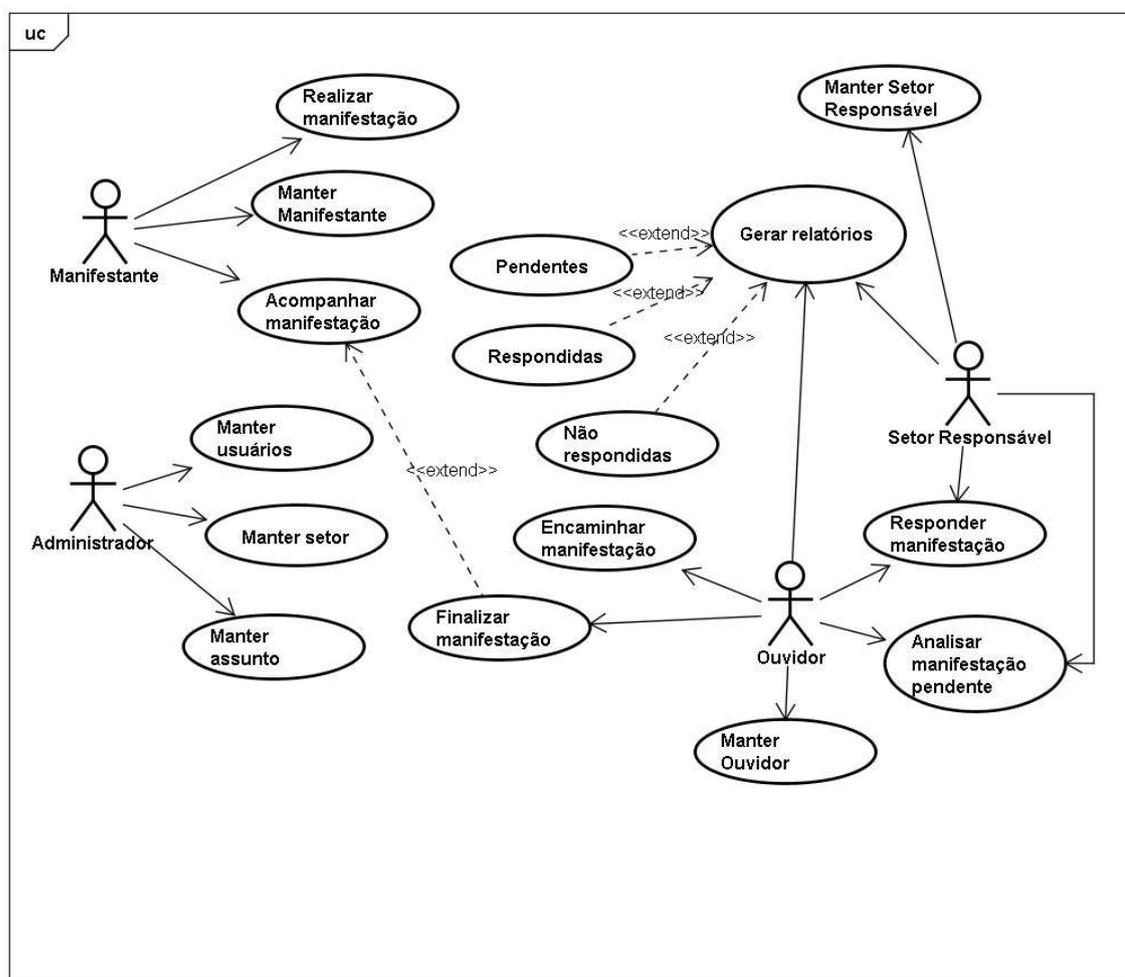


Figura 6 – Diagrama de casos de uso - Sistema de Ouvidoria

Neste diagrama, o ator **Manifestante** é a pessoa que irá acessar o sistema e registrar suas manifestações, ou seja, pode ser um cliente de uma empresa ou um cidadão quando o sistema for direcionado a uma organização pública. Ele também tem a permissão para alterar (manter) seus dados a qualquer momento.

Já o ator **Ouvidor**, é representado por um funcionário da organização, cuja função é analisar as manifestações, encaminhá-las para o setor responsável, e quando necessário responder diretamente uma manifestação. Também possui a permissão para alterar seus dados cadastrais e emitir relatórios.

O ator denominado **Setor Responsável**, é aquele que receberá as manifestações encaminhadas pelo **Ouvidor**. Sua função é analisar as manifestações pendentes e responder as manifestações dentro do prazo exigido. Este ator também pode emitir relatórios e alterar seus dados cadastrais.

O ator **Administrador** é o responsável por cadastrar usuários, assuntos e setores no sistema.

Os casos de uso representam as funções que o sistema de ouvidoria precisa oferecer aos usuários, possibilitando um bom funcionamento do sistema e fazendo com que as atividades sejam executadas de maneira correta.

5.2 – DESCRIÇÕES DOS CASOS DE USO DO SISTEMA

A seguir serão apresentados os casos de uso do sistema de ouvidoria, onde será feita uma descrição de como o serviço é executado. Logo depois será mostrado como o sistema realiza essa tarefa, e então haverá uma representação gráfica desse caso de uso com seu respectivo ator.

5.2.1 – Realizar manifestação

Realizar manifestação significa cadastrar uma nova reclamação ou sugestão no sistema de ouvidoria, onde posteriormente essa manifestação registrada será designada ao setor responsável. Para que a manifestação seja registrada, é necessário que o **Manifestante** preencha alguns campos para que futuramente ele possa acompanhar o status dessa manifestação registrada.

O **Manifestante** não é obrigado a preencher todo o cadastro com seus dados pessoais, optando por uma manifestação anônima, porém não terá o privilégio de ter acesso a manifestação a qualquer instante e de forma detalhada, somente recebendo seu protocolo por email.

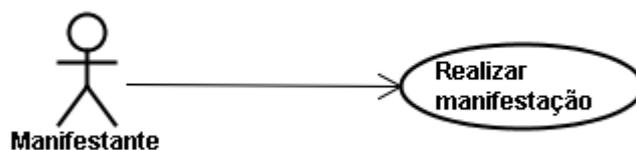


Figura 7 – Caso de uso realizar manifestação

5.2.2 – Acompanhar manifestação

Após o **Manifestante** registrar a sua manifestação, ele terá o privilégio de acompanhar o andamento da manifestação realizada, podendo optar a qualquer momento finalizar a manifestação caso tenha seu problema solucionado mesmo antes de obter uma resposta. Lembrando que caso o usuário opte por anonimato, seu acompanhamento será restringido a algumas funcionalidades.



Figura 8 – Caso de uso acompanhar manifestação

5.2.3 – Encaminhar manifestação

Durante o registro de uma manifestação, o manifestante escolhe o setor para o qual a sua manifestação deverá se encaminhada, e o **Ouidor** é o ator encarregado de analisar e encaminhar essa manifestação para o setor responsável.

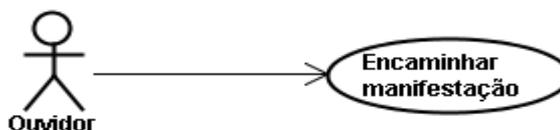


Figura 9 – Caso de uso encaminhar manifestação

5.2.4 – Analisar manifestação pendente (Ouvidor)

Após a manifestação ser encaminhada para o **Setor Responsável**, ela ficará com o status de pendente até que o **Setor Responsável** envie sua resposta para o Manifestante. Enquanto a manifestação não é respondida e encerrada, o **Ouvidor** tem o privilégio de analisar o andamento dessas manifestações. Após a manifestação ser encaminhada, o **Setor Responsável** também tem esse privilégio.



Figura 10 – Caso de uso analisar manifestação pendente

5.2.5 – Responder manifestação (Ouvidor)

Caso uma manifestação seja feita, porém não esteja de acordo com os padrões exigidos pelo sistema, o **Ouvidor** tem o privilégio de responder diretamente a manifestação sem precisar encaminhar para o setor responsável, onde essa resposta pode ser com a intenção de obter maiores detalhes, caso o manifestante não tenha conseguido expressar a sua necessidade, por exemplo.

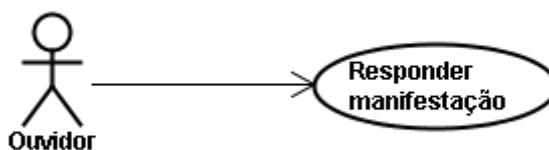


Figura 11 – Caso de uso responder manifestação

5.2.6 – Gerar relatórios

Os relatórios permitem que o **Ouvidor** e o **Setor responsável** tenham acesso as manifestações, de modo que possam obter um controle interno do que está ocorrendo durante um período determinado por eles, podendo consultar as manifestações pendentes, respondidas e as não respondidas.

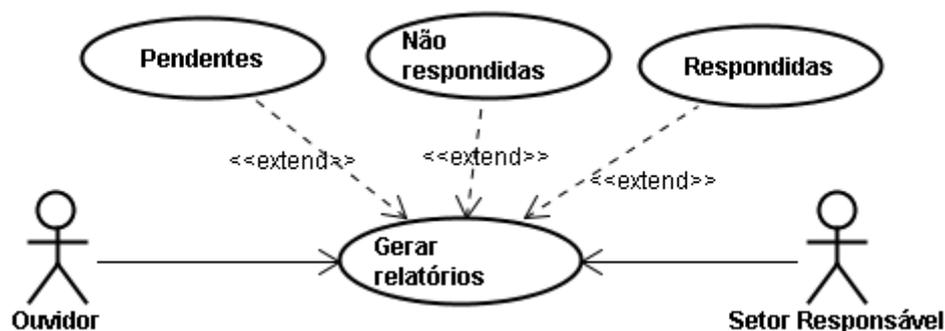


Figura 12 – Caso de uso gerar relatório

5.2.7 – Analisar pendentes

Enquanto o **Setor responsável** não responde a manifestação, ele pode analisar esta manifestação que se encontra pendente, para estudar as possíveis soluções para resolver o problema do manifestante.

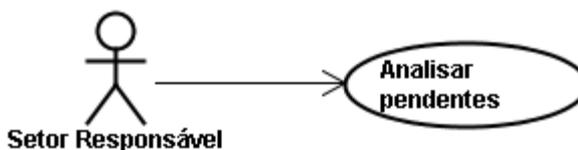


Figura 13 – Caso de uso analisar pendentes

5.2.8 – Responder manifestação (Setor Responsável)

Após o **Setor Responsável** receber o encaminhamento da manifestação, e também ter achado soluções para o possível problema solicitado, a resposta será enviada diretamente para o Manifestante, contendo informações que devem satisfazer a necessidade do manifestante. Caso contrário, essa resposta pode conter informações pedindo que seja necessário um novo contato com o Manifestante a fim de esclarecer maiores dúvidas.

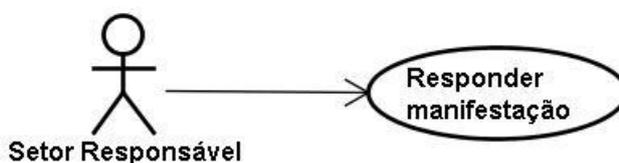


Figura 14 – Caso de uso responder manifestação (setor)

5.2.9 – Manter usuários

O **Administrador** é o responsável por cadastrar no sistema dois tipos de usuários: o Setor responsável e o Ouvidor. Depois de cadastrados, tais usuários poderão alterar seus dados conforme suas necessidades.

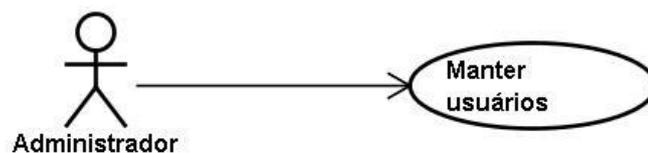


Figura 15 - Caso de uso Manter usuários

5.2.10 – Manter assunto

Ao surgir a necessidade de excluir, alterar ou cadastrar novos assuntos para atender a necessidade dos manifestantes, o **Administrador** é o responsável em realizar esta tarefa.

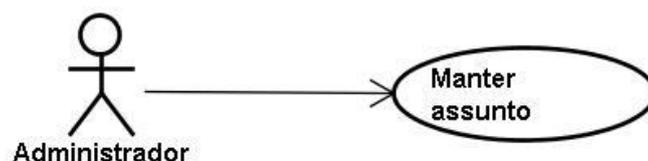


Figura 16 - Caso de uso manter assunto

5.2.11 – Manter setor

Pode acontecer de haver mudanças dentro da organização, onde sua estrutura pode ser modificada, incluindo novos setores e divisões. O **Administrador** também é o responsável por manter esses setores no sistema, podendo fazer inclusões, modificações e até mesmo excluir setores.

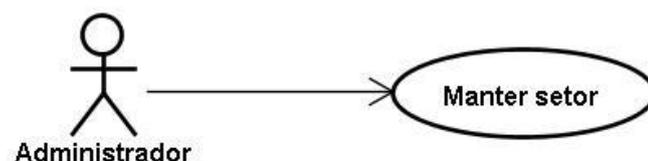


Figura 17 – Caso de uso manter setor

5.2.12 – Manter Manifestante

Caso de uso onde permite que o **Manifestante** realize alterações em seus dados cadastrais, incluindo gravar seus dados ao realizar uma manifestação.

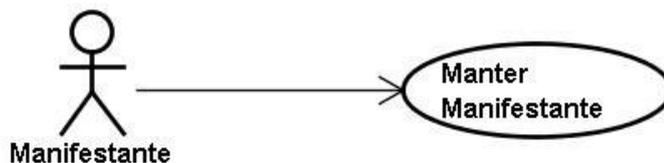


Figura 18 – Caso de uso manter Manifestante

5.2.13 – Manter Ouvidor

Caso de uso que permite com que o **Ouvidor** realize alterações em seus dados cadastrais.

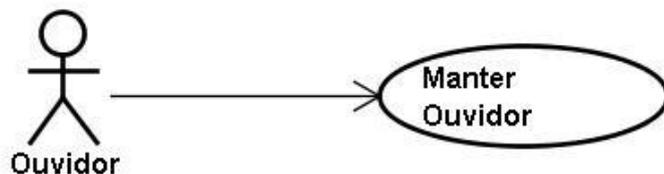


Figura 19 – Caso de uso manter Ouvidor

5.2.14 – Manter Setor Responsável

Este caso de uso dá permissão ao **Setor Responsável** realizar alterações em seus dados cadastrais.

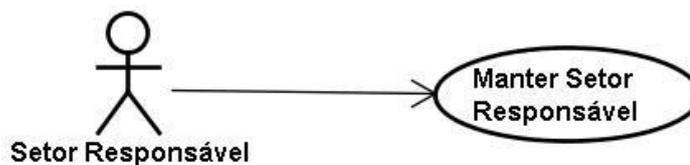


Figura 20 – Caso de uso manter Setor Responsável

5.3 – DIAGRAMA DE CLASSES – DEFINIÇÃO

Sendo um dos mais importantes dentre os diagramas, o diagrama de classes tem a funcionalidade de fazer com que todas as pessoas envolvidas no projeto possam compreender as funcionalidades do sistema, e também a sua estrutura.

Sua estrutura é formada basicamente por classes, atributos e métodos, servindo então como modelo para objetos. Os conceitos de objetos, classes, atributos e métodos foram explicados no capítulo 2 deste trabalho.

O diagrama de classes serve também para que o programador consiga enxergar de forma visual, todas as funcionalidades do sistema que será desenvolvido por ele, otimizando tempo no desenvolvimento, pois as classes utilizadas no projeto serão praticamente as mesmas utilizadas no desenvolvimento sistema.

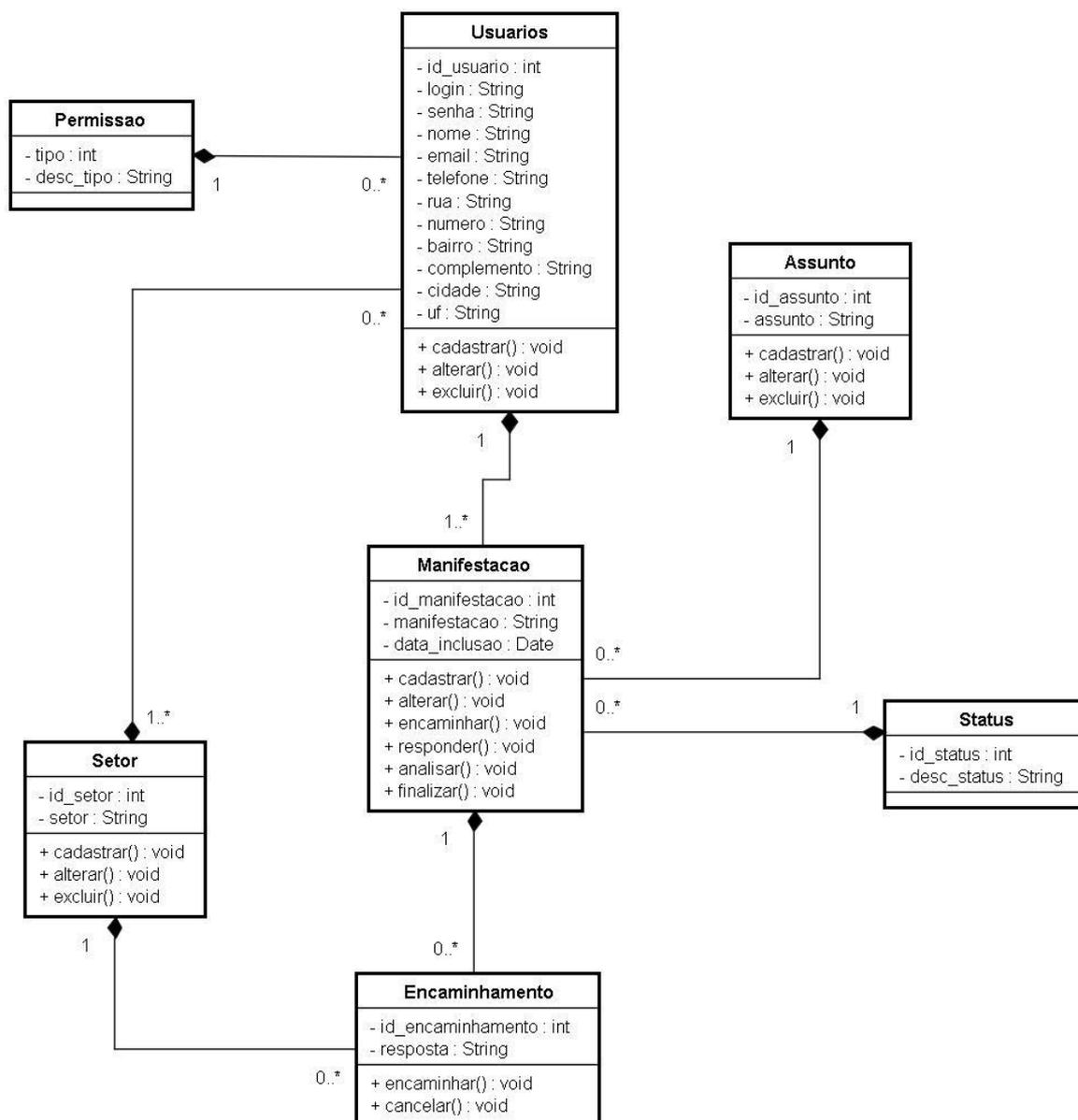


Figura 21 – Diagrama de classes do Sistema de Ouvidoria

5.4 – DIAGRAMA DE ENTIDADE E RELACIONAMENTO – DEFINIÇÃO

O diagrama de entidade e relacionamento (DER) é uma forma de representação gráfica que define a estrutura do conceito de Modelo Entidade Relacionamento.

Resumindo, o diagrama de entidade e relacionamento de forma geral a estrutura lógica do modelo do banco de dados, mostrando visualmente a relação entre as tabelas.

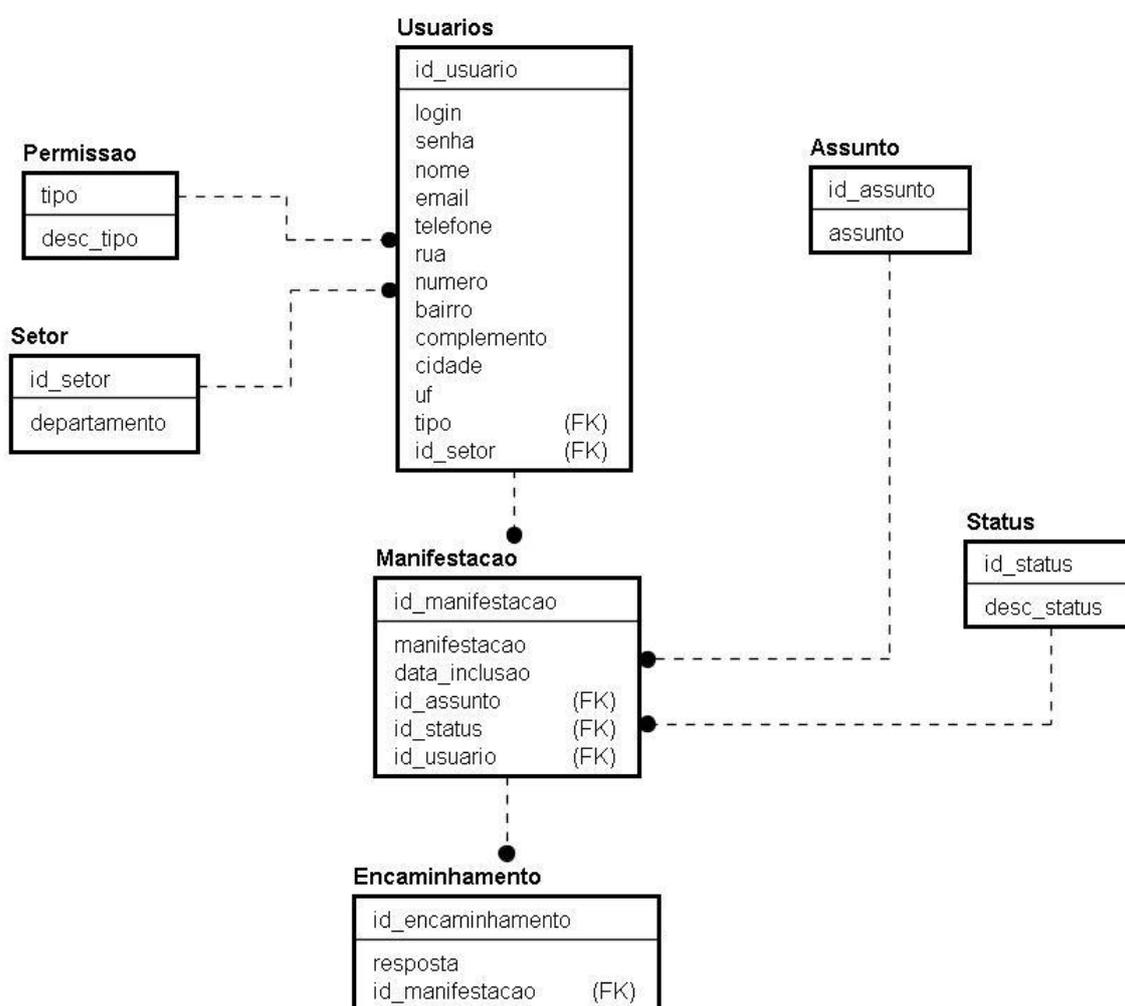


Figura 22 – Diagrama de entidade e relacionamento do Sistema de Ouvidoria

5.5 – DIAGRAMAS DE SEQUÊNCIA

5.5.1 – Diagrama de sequência – Realizar manifestação

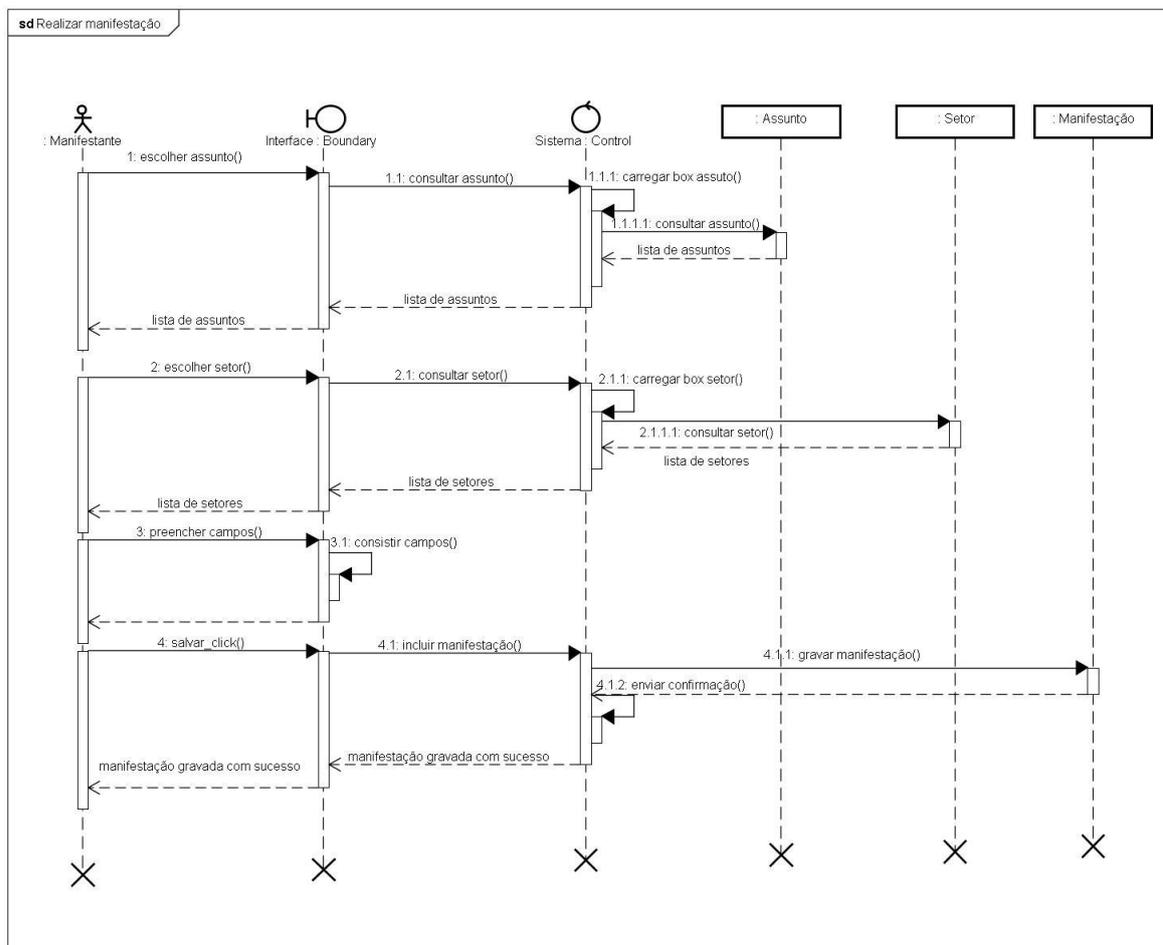


Figura 23 – Diagrama de sequência realizar manifestação

5.5.2 – Diagrama de seqüência – Acompanhar manifestação

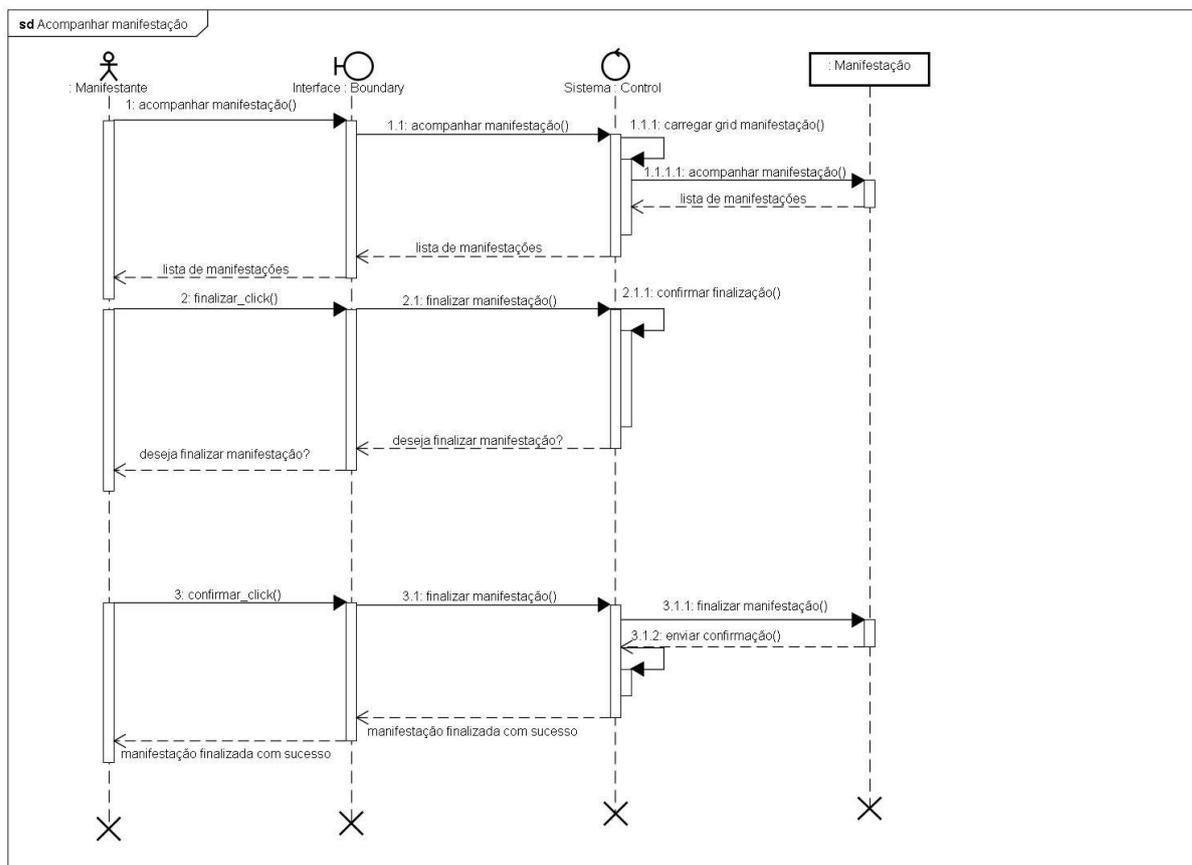


Figura 24 – Diagrama de seqüência acompanhar manifestação

5.5.3 – Diagrama de sequência – Encaminhar manifestação

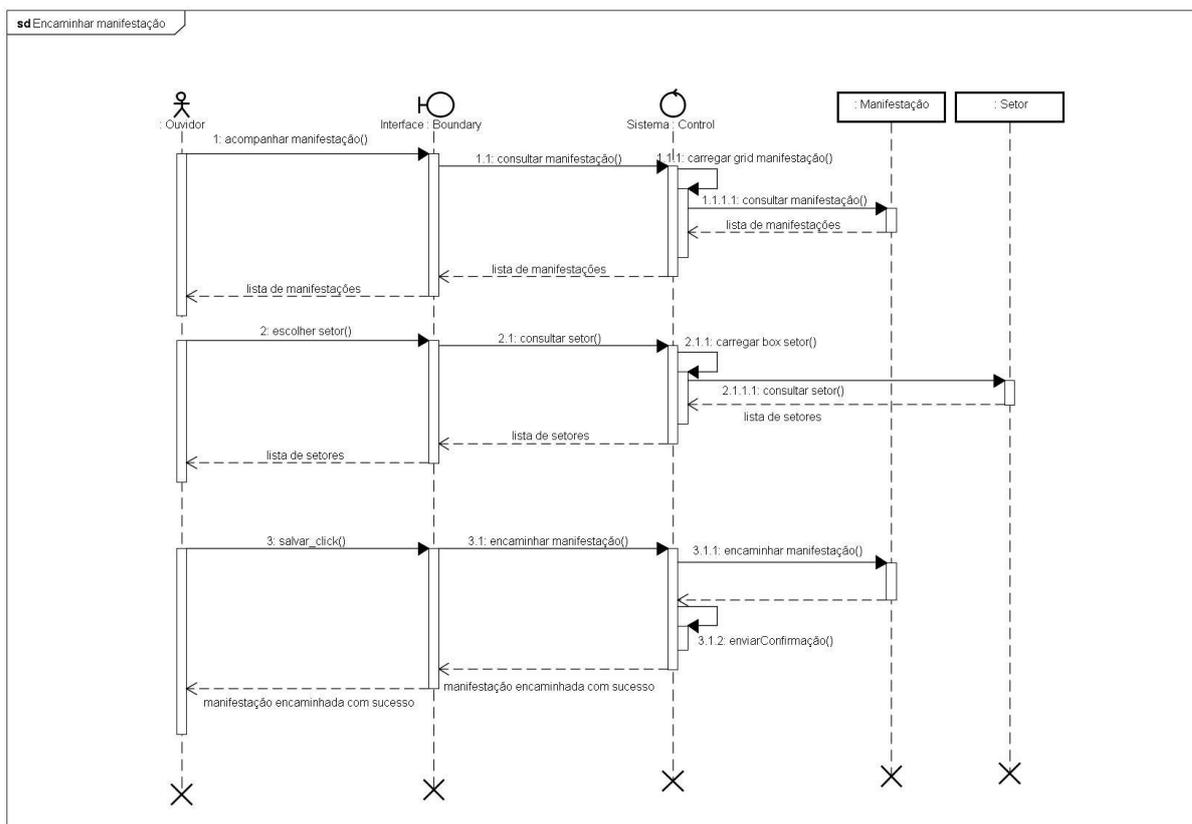


Figura 25 – Diagrama de sequência encaminhar manifestação

5.5.4 – Diagrama de seqüência - Analisar manifestação pendente (ouvidor)

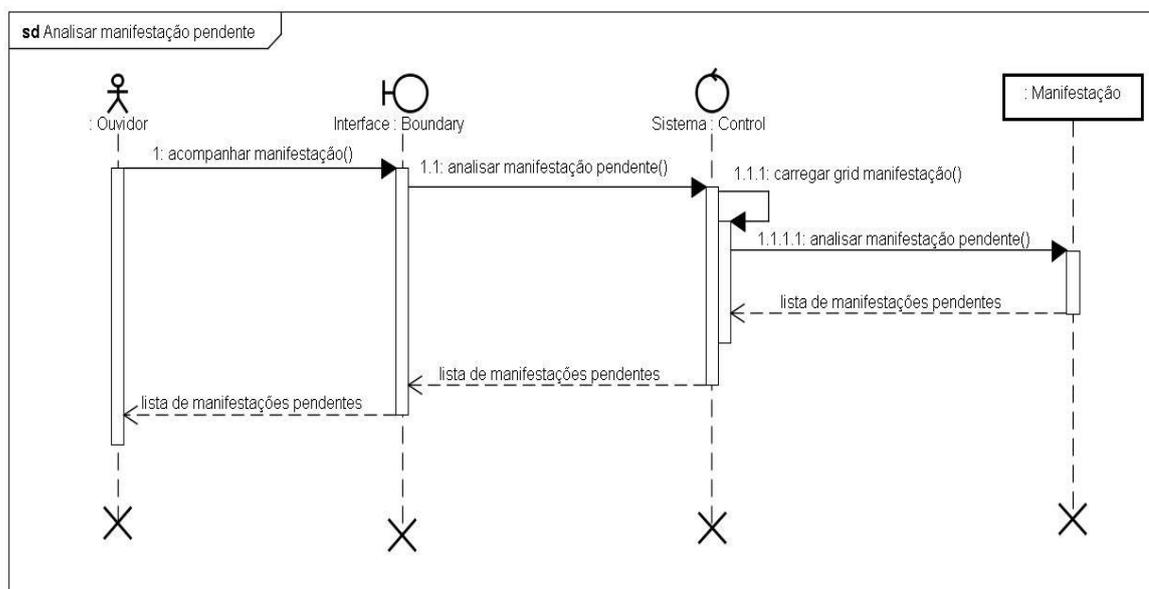


Figura 26 – Diagrama de seqüência analisar manifestação pendente (ouvidor)

5.5.5 – Diagrama de sequência – Responder manifestação (ouvidor)

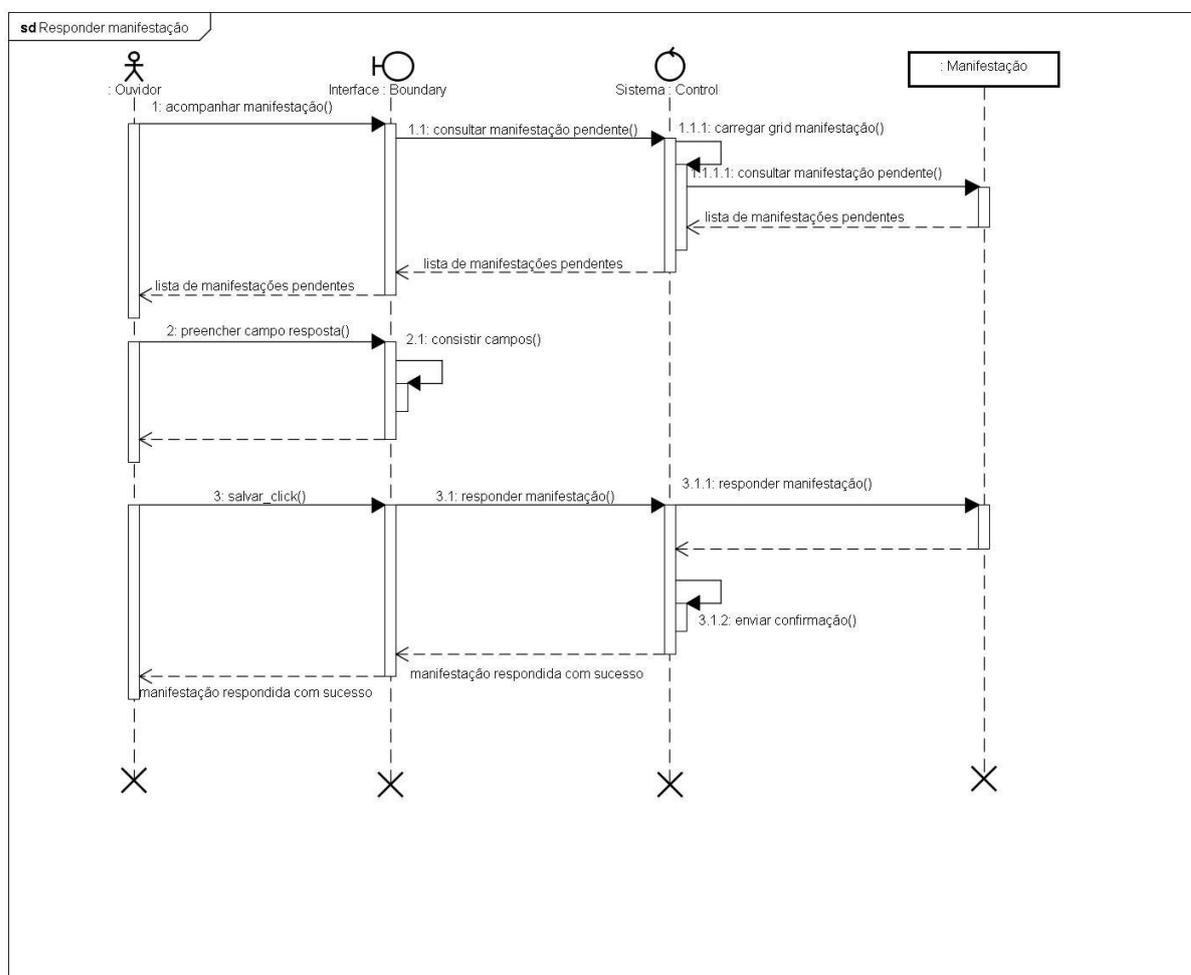


Figura 27 – Diagrama de sequência responder manifestação (Ouvidor)

5.5.6 – Diagrama de seqüência – Gerar relatórios

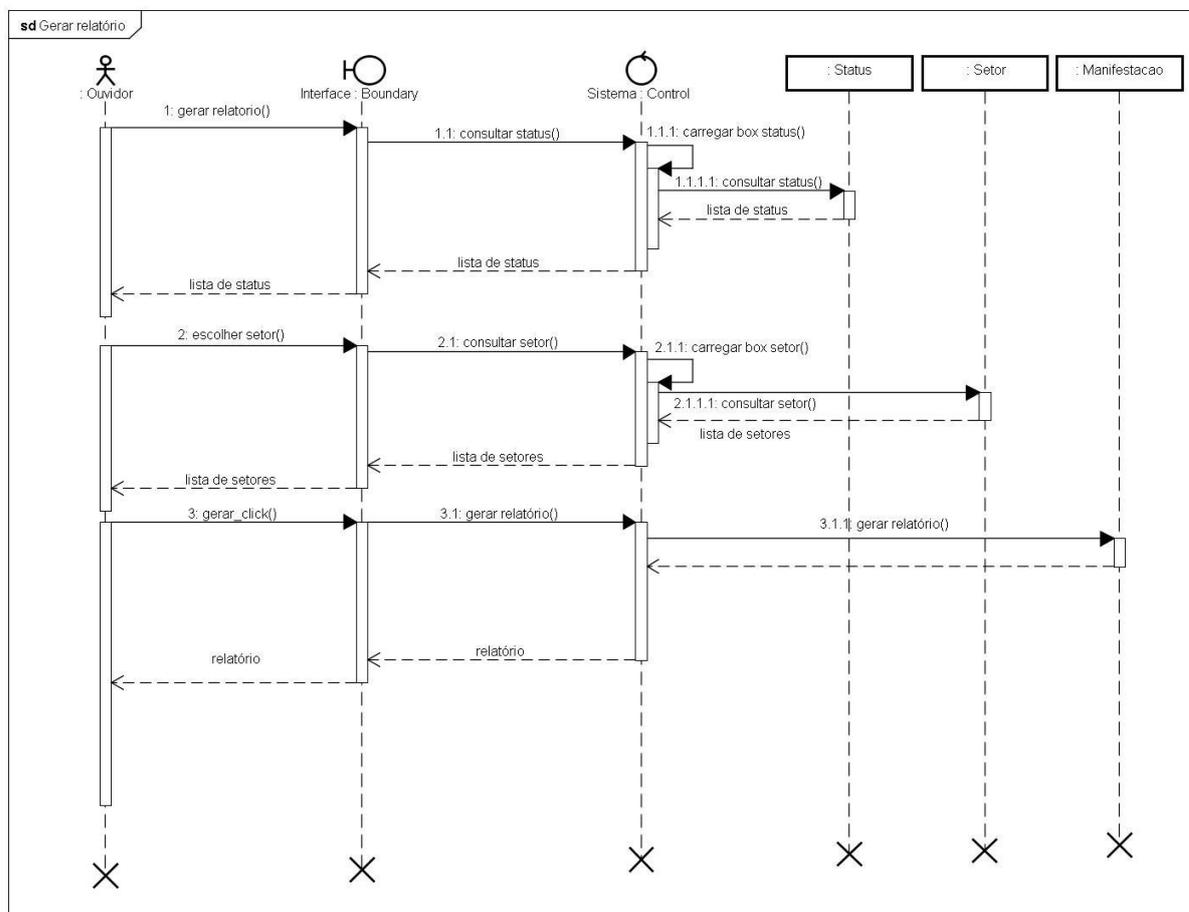


Figura 28 – Diagrama de seqüência gerar relatórios

5.5.7 – Diagrama de sequência – Analisar manifestações pendentes (setor responsável)

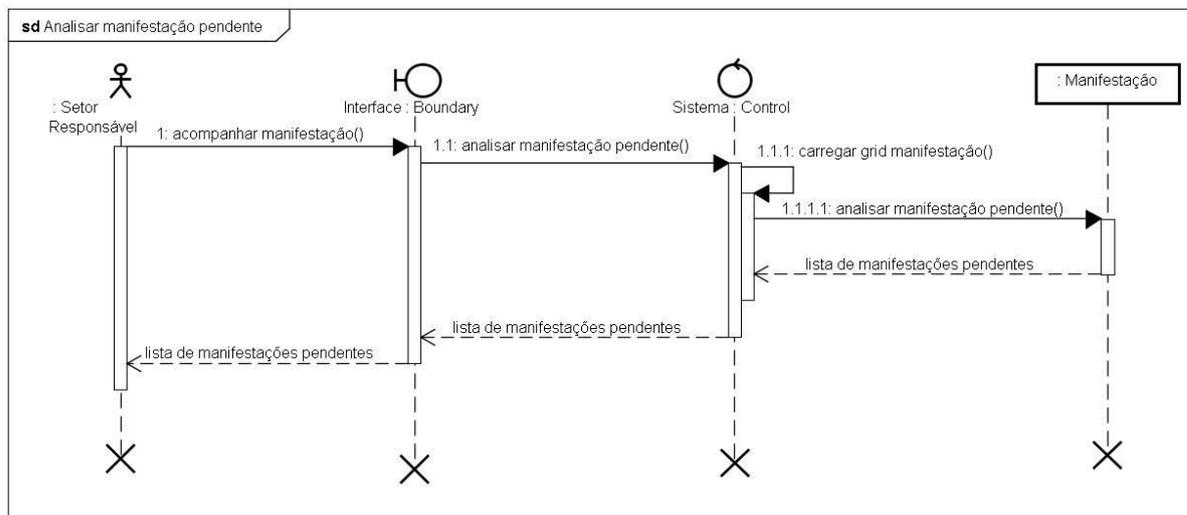


Figura 29 – Diagrama de sequência analisar manifestação pendente (setor responsável)

5.5.8 – Diagrama de sequência –Responder manifestação (setor responsável)

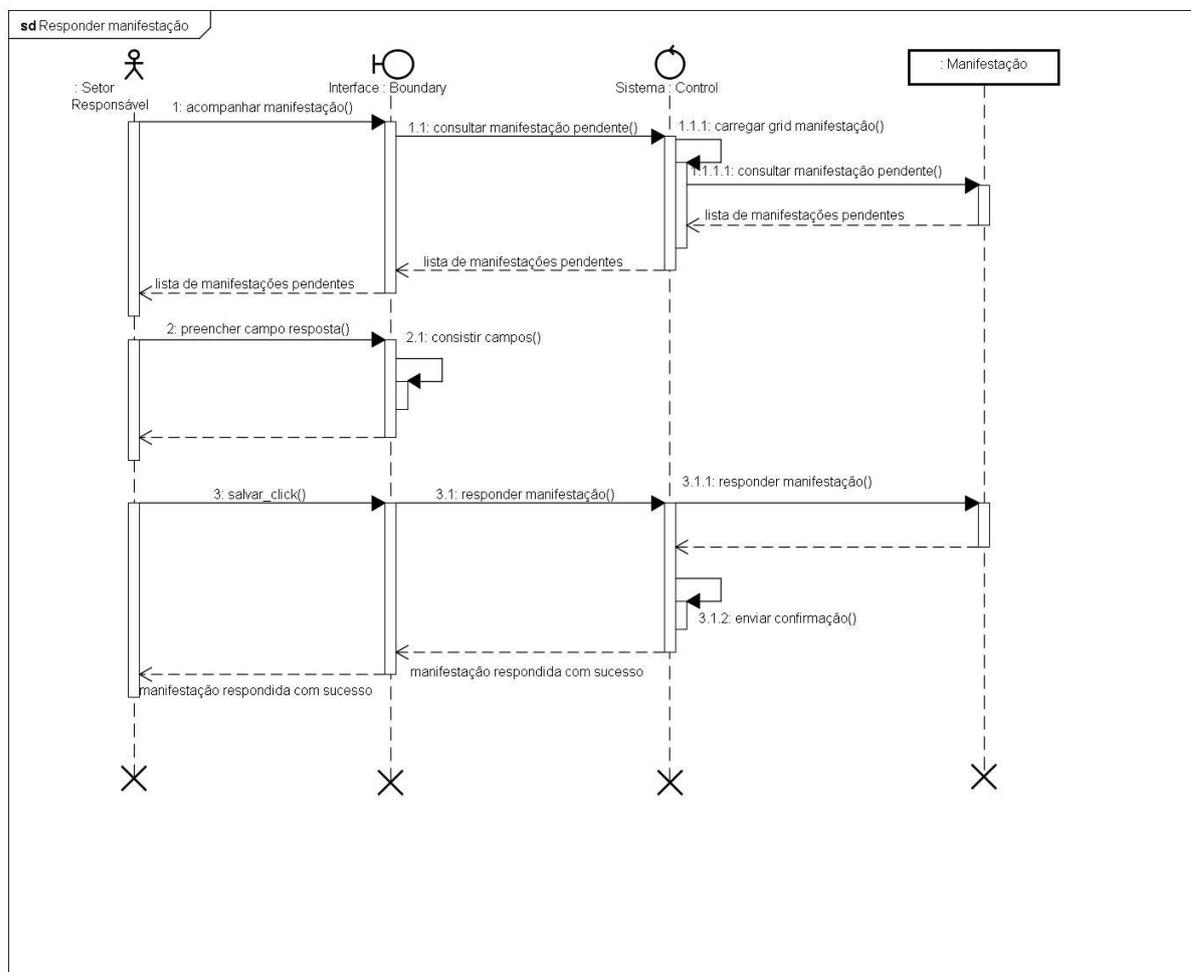


Figura 30 – Diagrama de sequência responder manifestação (setor responsável)

5.5.9 – Diagrama de seqüência – Manter usuários

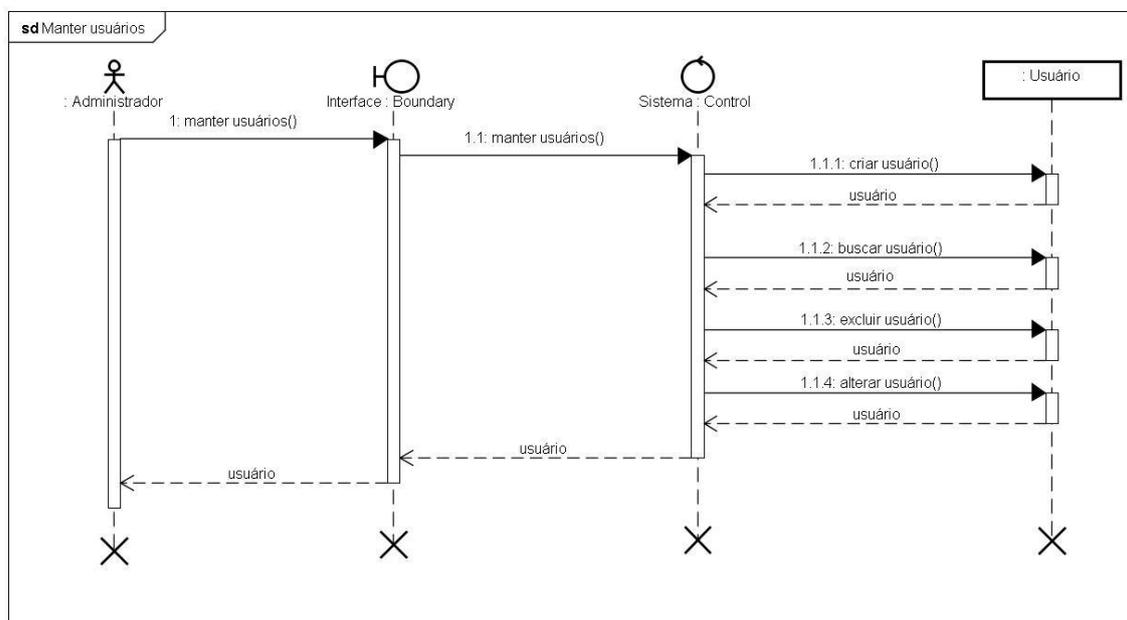


Figura 31 – Diagrama de seqüência manter usuários

5.5.10 – Diagrama de seqüência – Manter assunto

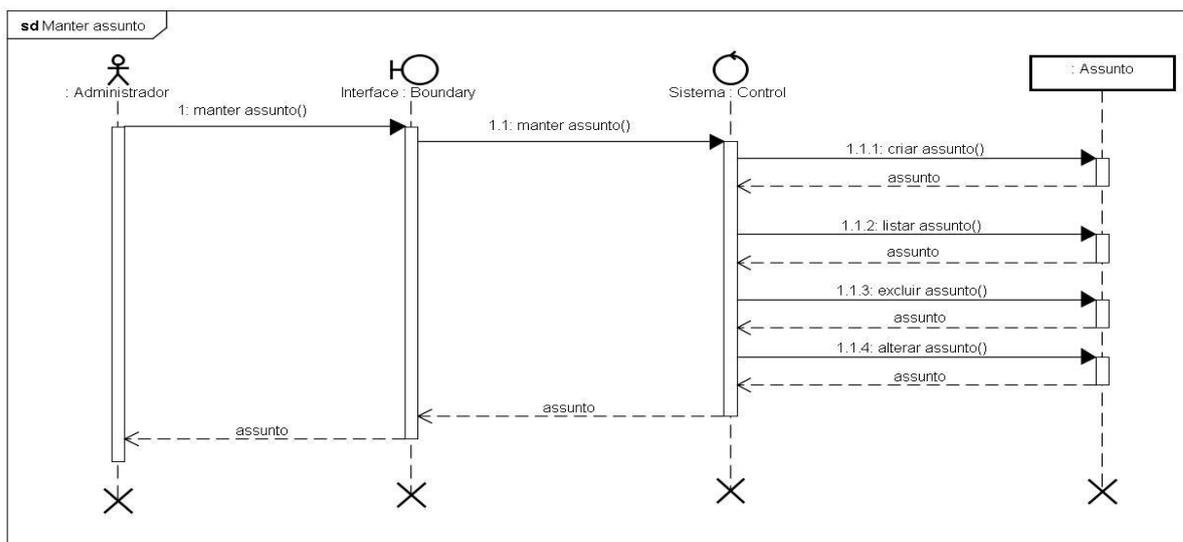


Figura 32 – Diagrama de seqüência manter assunto

5.5.11 – Diagrama de sequência – Manter setor

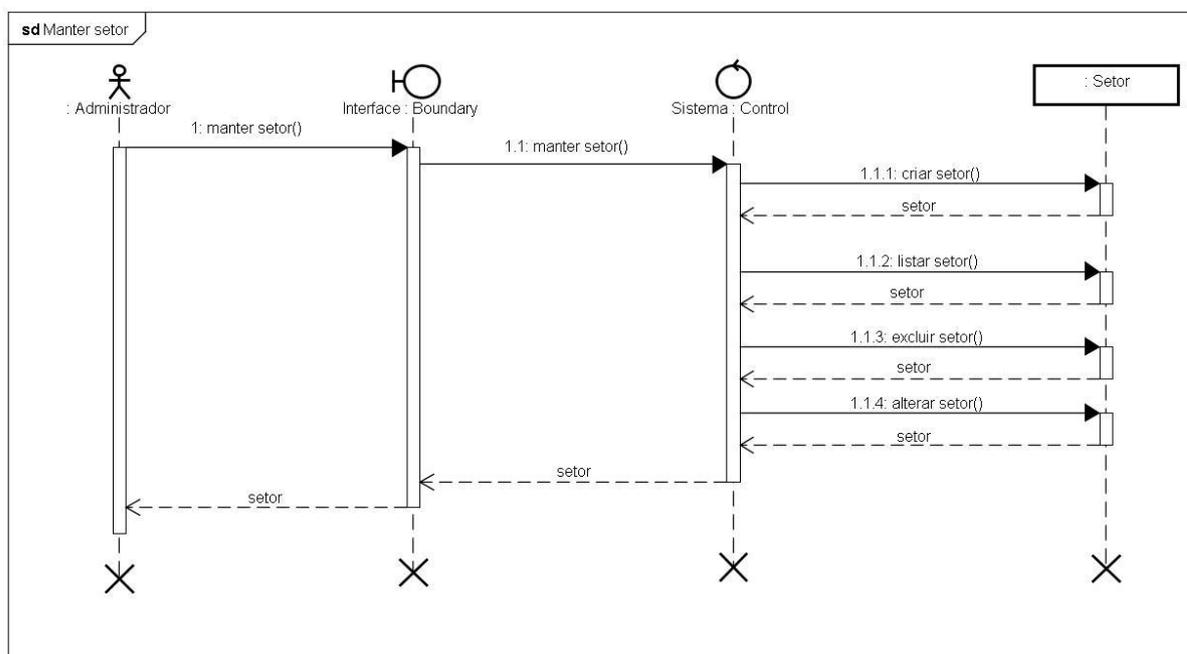


Figura 33 – Diagrama de sequência manter setor

5.5.12 – Diagrama de seqüência – Manter manifestante

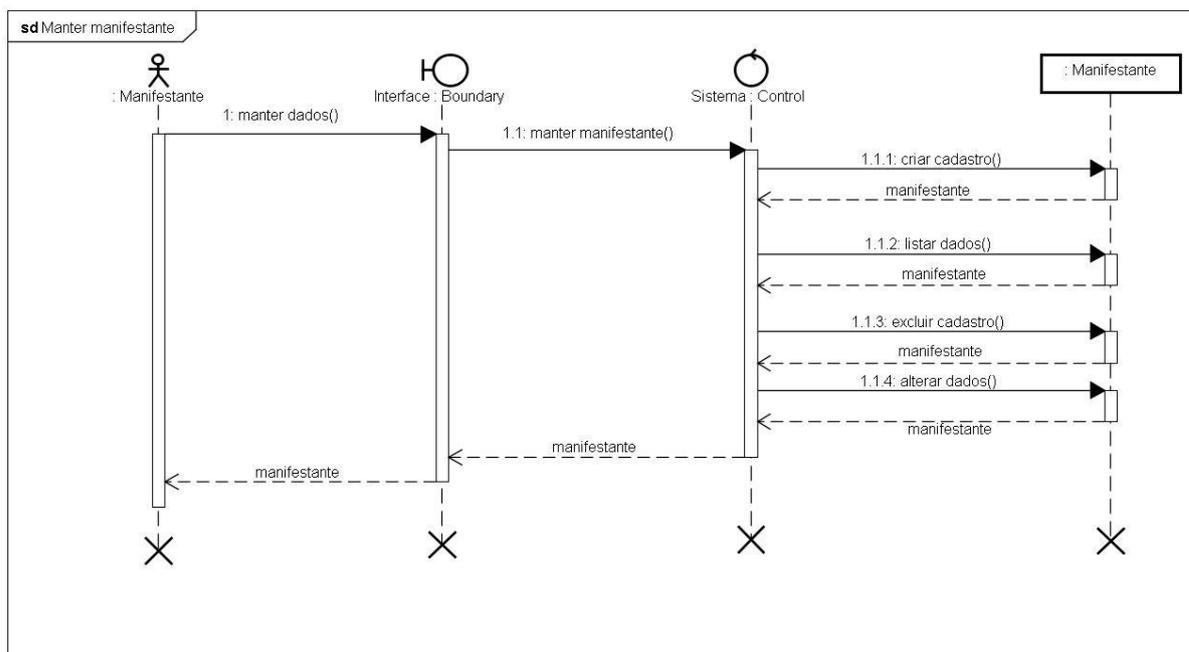


Figura 34 – Diagrama de seqüência manter manifestante

5.5.13 – Diagrama de seqüência – Manter ouvidor

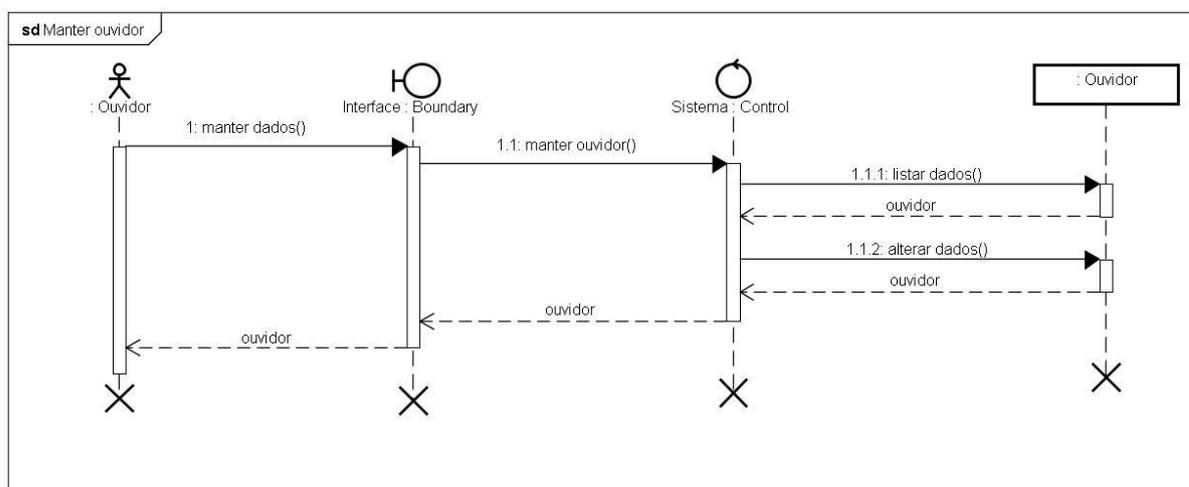


Figura 35 – Diagrama de seqüência manter ouvidor

5.5.14 – Diagrama de sequência – manter setor responsável

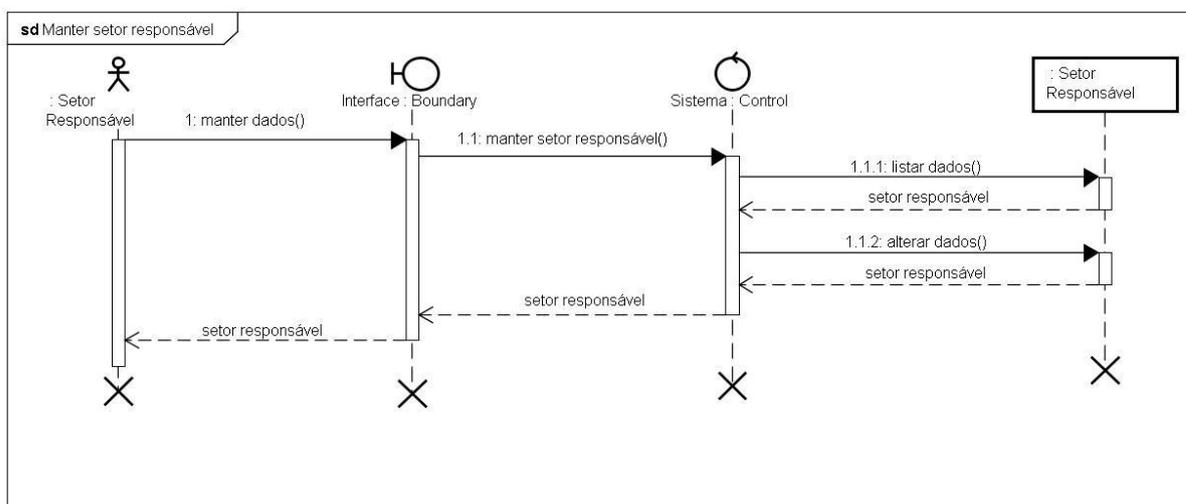


Figura 36 – Diagrama de sequência manter setor responsável

5.6 – DIAGRAMA DE ATIVIDADE – REALIZAR MANIFESTAÇÃO

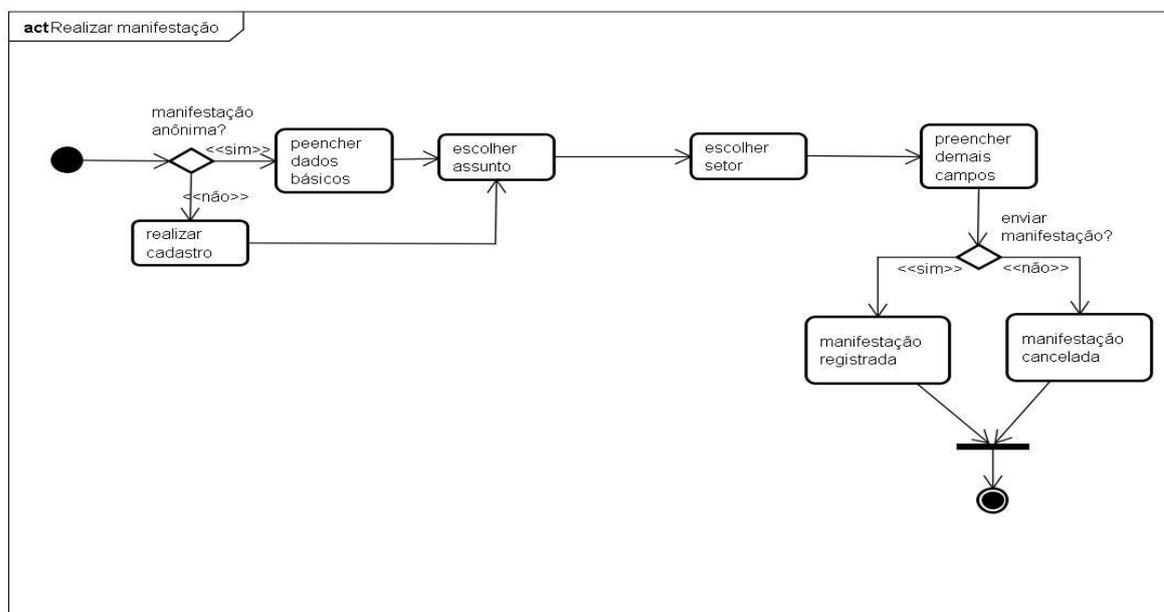


Figura 37 – Diagrama de atividade realizar manifestação

5.7 – CRONOGRAMA

	F E V	M A R	A B R	M A I	J U N	J U L	A G O	S E T	O T	N O V	D E Z
TAREFAS											
Levantamento de necessidades	X										
Especificação de Requisitos	X										
Levantamento de Requisitos		X									
Caso de Uso			X								
Especificação de Caso de Uso			X								
Diagrama de Classes				X							
Diagrama de Entidade e Relacionamento				X							
Diagrama de Sequência					X	X					
Diagrama de Atividades					X	X					
Programação							X	X	X	X	X
Testes									X	X	X
Implementação											X

Tabela 2 - Cronograma

CAPÍTULO 6 – IMPLEMENTAÇÃO

Conforme anteriormente mencionado, para a implementação do sistema foi utilizada a IDE Visual Studio 2012 Ultimate, com a linguagem de programação C# sobre a plataforma .NET.

6.1 – ORGANIZAÇÃO DO PROJETO

Para uma melhor organização das classes e pastas contidas no projeto, foi utilizado o conceito de camadas, onde cada camada tem sua função específica dentro do projeto. A seguir, a figura 25 mostra essas divisões do projeto e o conteúdo de cada camada.

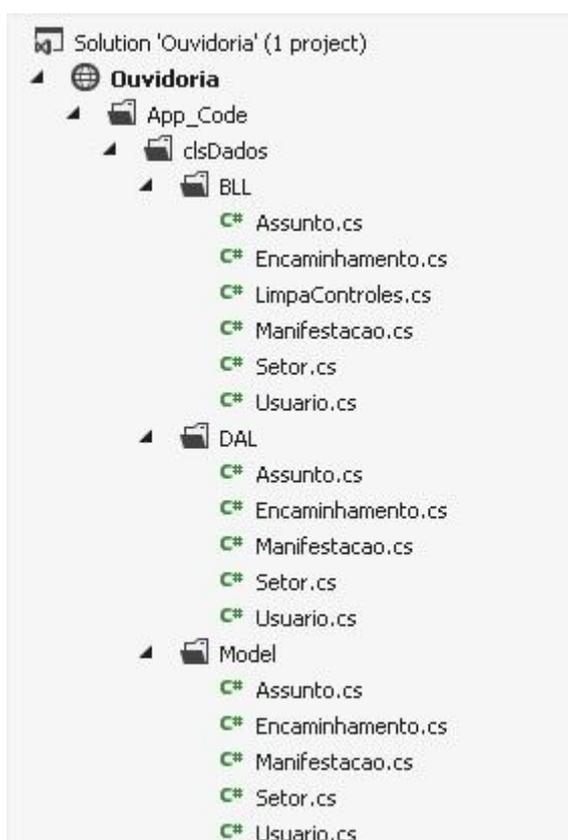


Figura 38 – Pastas de camadas do projeto e seus arquivos

BLL (Business Logic Layer): Camada responsável pela regra de negócio do sistema, ou seja, é responsável pela validação dos objetos. Os objetos não tem acesso direto ao bando de dados, tendo então que passar primeiramente pela BLL, e logo depois a BLL chama a camada DAL para acessar o banco de dados.

DAL (Data Access Layer): Camada responsável pela conexão e interação com o banco de dados. É nessa camada que deve ser configurado o endereço de acesso ao banco de dados, e também programação responsável pelas alterações, como incluir, alterar, excluir e ler os dados.

Model (Modelo): Por sua vez, a camada Model é responsável por conter as classes encapsuladas de dados, guardando os objetos e seus tipos, de forma que deverão ser utilizadas para movimentar esses objetos entre as camadas DAL e BLL.

6.2 – INTERFACE

A seguir serão apresentadas as principais telas do Sistema de Ouvidoria

6.2.1 – Tela de início

Ao acessar o Sistema de Ouvidoria, o usuário se depara com a página de início, onde poderá realizar *login* quando administrador, ou consultar e registrar manifestações quando for um cidadão.

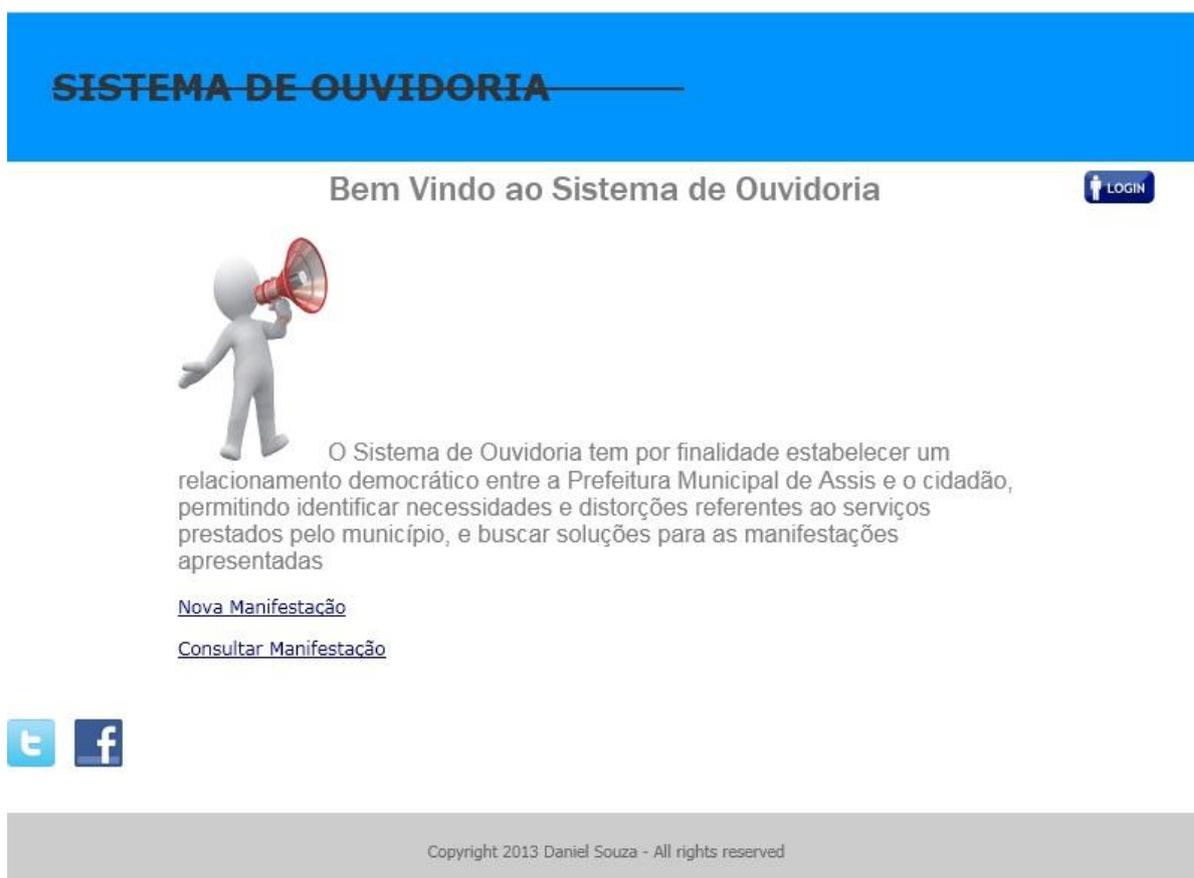
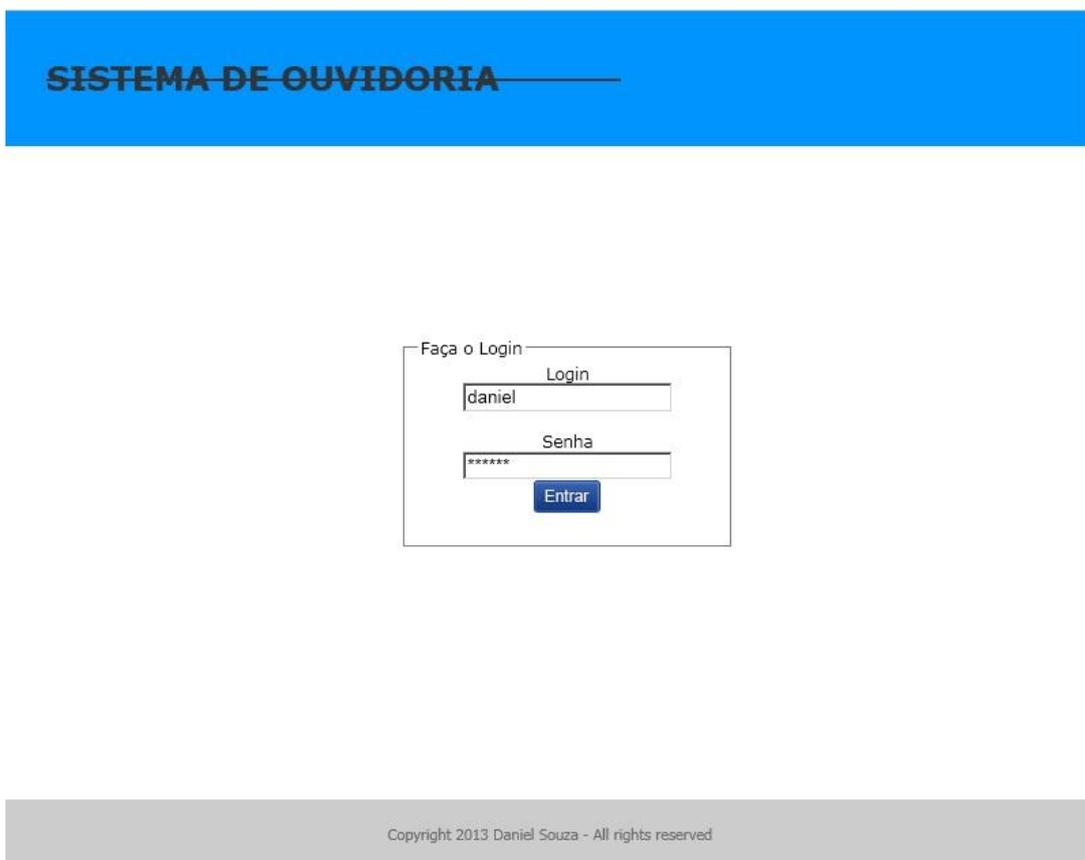


Figura 39 – Tela de início do Sistema de Ouvidoria

6.2.2 – Tela de login

Tela de login, onde o Administrador, o Ouvidor e o Setor Responsável irão acessar para entrar no sistema.



The image shows a web application interface for a system titled "SISTEMA DE OUVIDORIA". The page has a blue header with the title in white. Below the header, there is a central login form titled "Faça o Login". The form contains two input fields: "Login" with the text "daniel" and "Senha" with asterisks. Below the fields is a blue "Entrar" button. At the bottom of the page, there is a grey footer with the text "Copyright 2013 Daniel Souza - All rights reserved".

Figura 40 – Tela de login do sistema

6.2.3 – Tela manter usuário

Tela onde os usuários poderão realizar alterações em seus dados cadastrais.

SISTEMA DE OUVIDORIA

Home Meus dados

Dados Cadastrais.:

Nome

Telefone

E-mail

Endereço.:

Rua Nº

Bairro Complemento

Cidade UF

[Alterar senha](#)

Copyright 2013 Daniel Souza - All rights reserved

Figura 41 – Tela manter usuário

6.2.4 – Tela registrar manifestação

Nesta tela o usuário escolhe o assunto referente a sua manifestação, e escolhe também o setor para o qual a manifestação deverá ser encaminhada. Após escolher o assunto, o setor, e redigir sua manifestação no campo solicitado, o usuário clica no botão enviar para registrar sua manifestação.

SISTEMA DE OUVIDORIA

Registre a sua manifestação

O sistema permite que as informações enviadas sejam armazenadas de forma segura e sigilosa e são essenciais para o devido encaminhamento das demandas e demais comunicações junto aos órgãos responsáveis facilitando um retorno satisfatório das questões apresentadas.

Escolha o assunto

Reclamação

Escolha o setor

Obras

Digite aqui a sua manifestação

Boa tarde
Eu queria fazer uma reclamação a respeito do asfalto aqui do meu bairro, pois são muitos os buracos encontrados em quase todas as ruas... Moro no Jd Paraná, e gostaria de uma posição a respeito da situação
Obrigado

Cancelar

Enviar

Figura 42 - Tela registrar manifestação

6.2.5 – Tela acompanhar manifestação

Ao entrar na tela inicial do sistema, o usuário se depara com um link escrito “consultar manifestação”, onde será redirecionado para a tela para realizar a consulta conforme a figura 30.

A consulta pode ser feita de duas maneiras: realizando login ou anonimamente. Quando o usuário escolhe consultar realizando login, ele também poderá alterar seus dados cadastrais, analisar mais de uma manifestação, acompanhando individualmente o status de cada uma. Já o usuário anônimo não tem esse privilégio, podendo apenas acompanhar o status atual da manifestação.

Para consultar uma manifestação anônima, o usuário deverá preencher o campo com o número do protocolo, e também confirmar sua autenticidade preenchendo o campo captcha conforme as letras mostradas na imagem.

SISTEMA DE OUVIDORIA

Login

Consultar manifestação anônima

Digite o número do protocolo

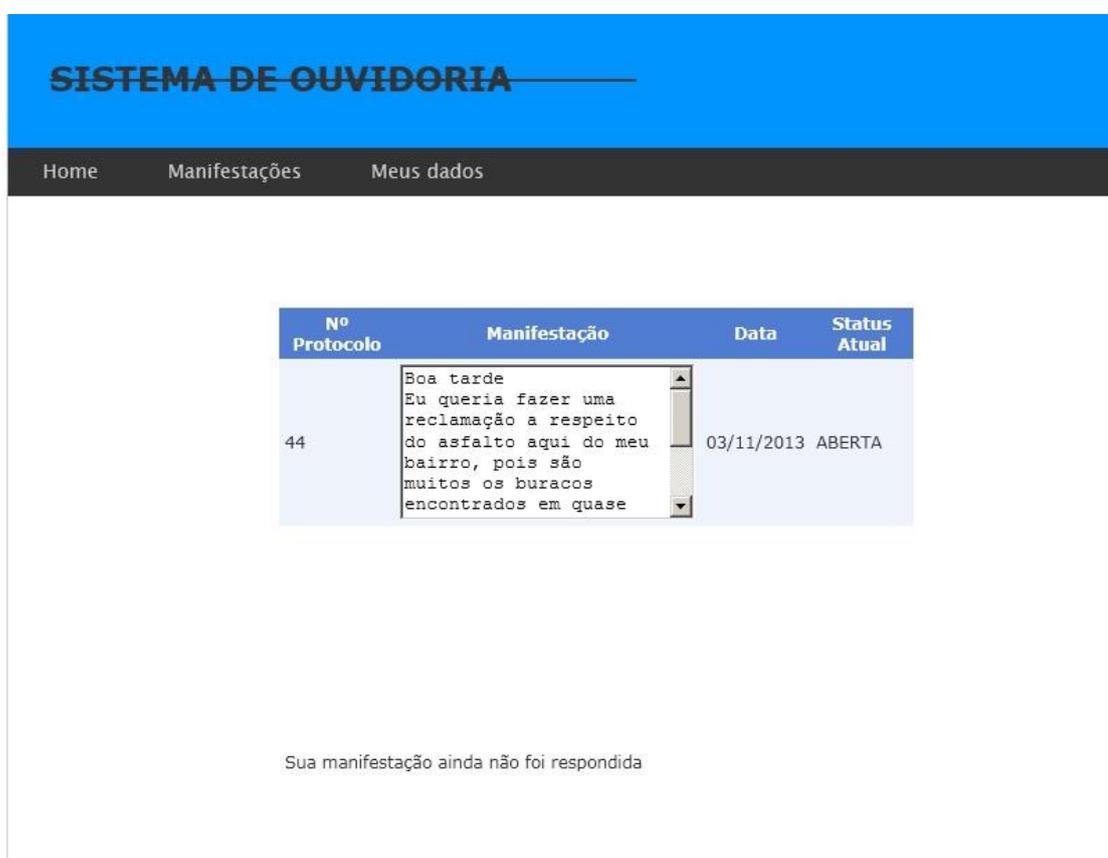


Digite os numeros da imagem

Figura 43 – Tela acompanhar manifestação

6.2.6 – Tela status da manifestação

A tela status da manifestação apresenta o status atual em que a mesma se encontra, mostrando o numero do protocolo, a data em que foi registrada, a descrição da manifestação realizada, e quando respondida, irá exibir na parte inferior da tela uma nova tabela contendo a resposta. Caso ainda não tenha sido respondida, será exibida a mensagem “Sua manifestação ainda não foi respondida”.



The screenshot shows a web interface for a complaint system. At the top, there is a blue header with the text "SISTEMA DE OUVIDORIA". Below the header is a navigation bar with three items: "Home", "Manifestações", and "Meus dados". The main content area displays a table with the following data:

Nº Protocolo	Manifestação	Data	Status Atual
44	Boa tarde Eu queria fazer uma reclamação a respeito do asfalto aqui do meu bairro, pois são muitos os buracos encontrados em quase	03/11/2013	ABERTA

Below the table, the message "Sua manifestação ainda não foi respondida" is displayed.

Figura 44 – Tela status da manifestação

CONSIDERAÇÕES FINAIS

O Capítulo 1 apontou a grande necessidade que a cidade de Assis tem de implantar um Sistema de Ouvidoria, devido ao enorme número de reclamações presenciadas através das redes sociais e TV na própria cidade, e também em outras cidades da região.

Ao ser implementado o Sistema de Ouvidoria na cidade, os setores responsáveis por administrar os serviços da cidade terão sua informações centralizadas em um único lugar, podendo então se dedicar mais em atender as pessoas através do canal de Ouvidoria, fazendo com que os cidadãos tenham uma maior satisfação e segurança em entrar em contato com a Prefeitura.

Por meio das redes sociais, televisão, e entrevistas com funcionários públicos, foi possível identificar as funcionalidade do sistema, identificando as principais necessidades, onde através de diagramas, foi possível obter uma visão detalhada de como funcionaria cada divisão do Sistema de Ouvidoria.

Apesar de o sistema atender todas as necessidades que um Sistema de Ouvidoria necessite, há programações possíveis de se adicionar ao sistema, a fim de melhorar a qualidade do software e reforçar a segurança do mesmo, porém o objetivo principal do sistema foi alcançado como citado acima, podendo a qualquer momento ser utilizado pelo município.

REFERÊNCIAS BIBLIOGRÁFICAS

BATTISTI, Júlio. **ASP.NET: Uma Revolução na Construção de Sites e Aplicações Web**. Júlio Battisti Editora, 2006.

BERTALANFFY, Ludwig Von. **Teoria Geral dos Sistemas**. Editora Vozes, 1975.

BORATTI, Isaias Camilo. **Programação Orientada a Objetos Delphi**. Editora Visual Books, 2002.

GUEDES, Gilleanes t.a.. **UML 2 uma abordagem prática**. Editora Novatec, 2ª edição, 2011.

MELO, Décio Heinzelmann Luckow Alexandre Altair de. **Programação Java para Web**. Novatec Editora Ltda, 2010.

PRESSMAN, Roger S.. **Engenharia de software: uma abordagem profissional**. Editora AMGH, 7ª edição, 2011.

ROCHA, Álvaro. **O Essencial da Análise de Sistemas**. UFP, 2008.

SIVA, Nelson Peres da. **Projeto e desenvolvimento de sistemas**. Editora Ética, 2001.

TEOREY, Toby, LIGHTTONE, Sam, NADEAU, Tom. **Projeto e modelagem de banco de dados**. Editora Campus, 2007.

VELLOSO, F. C.. **Informática: Conceitos Básicos**. Editora Campus, 2004.

Referências Eletrônicas

Site do Microsoft Visual Studio < <http://msdn.microsoft.com/pt-br/vstudio/> > acessado em 12 de outubro de 2013.

Site do .NET Framework < <http://msdn.microsoft.com/pt-br/vstudio/aa496123> > acessado em 12 de outubro de 2013.

Site do Microsoft SQL Server < <http://www.microsoft.com/sqlserver/pt/br/product-info.aspx> > acessado em 12 de outubro de 2013.

Site do Microsoft SQL Management Studio < <http://www.microsoft.com/pt-br/download/details.aspx?id=7593> > acessado em 12 de outubro de 2013.