

DANIEL DE FILIPPO LIMA

DESENVOLVIMENTO DE SOFTWARE PARA WEB USANDO A  
PLATAFORMA JEE, PARA UMA APLICAÇÃO DE OUVIDORIA. NA  
ADMINISTRAÇÃO PÚBLICA.

2011

DESENVOLVIMENTO DE SOFTWARE PARA WEB USANDO A  
PLATAFORMA JEE, PARA UMA APLICAÇÃO DE OUVIDORIA. NA  
ADMINISTRAÇÃO PÚBLICA.

DANIEL DE FILIPPO LIMA

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

Orientador: \_\_\_\_\_

Analisador (1): \_\_\_\_\_

Analisador (2): \_\_\_\_\_

## FICHA CATALOGRÁFICA

DE FILIPPO, Daniel

Desenvolvimento de software para web usando a plataforma jee,  
para uma aplicação de ouvidoria. Na administração pública.

Daniel De Filippo Lima. Fundação Educacional do Município de Assis – Fema :

Assis, 2011

57p.

Orientador: Douglas Sanches da Cunha

Trabalho de Conclusão de Curso (TCC) – Tecnologia em Processamento de  
Dados - Instituto Municipal de Ensino Superior de Assis.

CDD. 004.68

## AGRADECIMENTOS

Agradeço a primeiramente á Deus que me capacitou por chegar até aqui, dando-me força, saúde e sabedoria.

Ao meu orientador Ms. Douglas Sanches da Cunha, pelas suas orientações, ideias, opiniões e por sempre me incentivar durante este trabalho.

Agradeço aos meus Pais, por terem acreditado na minha capacidade, sempre estiveram me ajudando e apoiando em todos os momentos.

## RESUMO

Ouvidoria é um serviço oferecido à população para receber críticas e sugestões sobre o desempenho de órgãos públicos ou empresas privadas. Atua no fortalecimento da cidadania.

O software é uma aplicação desenvolvida para a web, feito em Java juntamente com o framework JavaServer Pages, possibilita que seja feito um serviço com mais transparência, agilidade e segurança.

**Palavras - chave:** Java, JavaServer Pages.

## ABSTRACT

Ombudsman is a service offered to the public to receive comments and suggestions on the performance of public agencies or private companies. It operates in the strengthening of citizenship.

The software is an application developed for the web, written in Java with the JavaServer Pages, provides a service that is made with more transparency, agility and safety.

**Keywords:** Java, JavaServer Pages.

## LISTA DE ABREVIATURAS E SIGLAS

UML	Unified Modeling Language
SQL	Structured Query Language
MVC	Model View Controller
UI	User Interface
JSP	Java Server Pages
IDE	Integrated Development Environment
API	Application Program Interface
ORM	Object-Relational Mapping
XML	Extensible Markup Language
WBS	Work Breakdown Structure

## LISTA DE ILUSTRAÇÕES

Figura 1 Diagrama WBS.....	22
Figura 2 Sequenciamento das Atividades Definidas.....	24
Figura 3 Diagrama de Classe.....	30
Figura 4 Diagrama De Entidade Relacionamento.....	31
Figura 5 Diagrama Caso de Uso Global.....	32
Figura 6 Use Case Cadastrar Ouvidor.....	33
Figura 7 Use Case Alterar Ouvidor.....	34
Figura 8 Use Case Excluir Ouvidor.....	35
Figura 9 Listar todos os Ouvidores.....	36
Figura 10 Listar todos os Usuários.....	37
Figura 11 Listar todas as Pessoas Físicas.....	38
Figura 12 Listar todas as Pessoas Jurídicas.....	39
Figura 13 Listar Todos os Anônimo.....	40
Figura 14 Listar todas as Manifestações.....	41
Figura 15 Listar Mensagens dos Ouvidores para alguma Manifestação.....	42
Figura 16 Listar todas as Mensagens de algum Ouvidor.....	43
Figura 17 Responder Manifestação.....	44
Figura 18 Listar Todas as Manifestações.....	45
Figura 19 Responder Manifestação para o Administrador.....	46
Figura 20 Fazer Manifestação.....	47
Figura 21 Use Case Acompanhar Manifestação.....	48
Figura 22 Diagrama de Sequência Administrador.....	49
Figura 23 Diagrama de Sequência Ouvidor.....	50
Figura 24 Diagrama de Sequência Usuário.....	51
Figura 25 Tela Inicial.....	54
Figura 26 Tela Fazer Manifestação.....	55
Figura 27 Tela Ouvidor.....	56
Figura 28 Tela Administrador.....	57

## LISTA DE TABELAS

Tabela 01 - Orçamento do Projeto.....	27
Tabela 02 – Cronograma das Atividades Definidas.....	28

# SUMÁRIO

<b>1. INTRODUÇÃO</b>	<b>11</b>
1.2 OBJETIVO	12
1.3 JUSTIFICATIVA	12
1.4 PUBLICO ALVO	12
<b>2. LEVANTAMENTO DOS REQUISITOS</b>	<b>13</b>
2.1 FORMA DE LEVANTAMENTO ADOTADA	13
2.2 RESULTADOS ESPERADOS NA IMPLANTAÇÃO DO SOFTWARE	13
2.3 DETALHAMENTO DO PROBLEMA A SER RESOLVIDO	13
<b>3. ANÁLISE DOS REQUISITOS</b>	<b>14</b>
3.1 PROBLEMAS POTENCIAIS	14
3.2 RESTRIÇÕES DO SISTEMA	14
<b>4. ESPECIFICAÇÃO DOS REQUISITOS</b>	<b>15</b>
4.1 ESCOPO	15
<b>5. MÉTODO DESENVOLVIMENTO</b>	<b>16</b>
5.1 AMBIENTES DE DESENVOLVIMENTO	16
<b>5.1.1 JAVA</b>	<b>16</b>
<b>5.1.2 DREAMWEAVER</b>	<b>17</b>
<b>5.1.3 JAVASERVER PAGES (JSP)</b>	<b>17</b>
5.2 FERRAMENTA PARA ANÁLISE	18
5.3 BANCO DE DADOS	21
<b>5.3.1 FERRAMENTA PARA GERENCIAR O BANCO DE DADOS</b>	<b>21</b>
5.4 WBS – WORK BREAKDOWN STRUCTURE	22
<b>5.4.1 DEFINIÇÕES DAS ATIVIDADES</b>	<b>23</b>
5.5 SEQUENCIAMENTO DAS ATIVIDADES DEFINIDAS	24
5.6 RECURSOS NECESSÁRIOS PARA O DESENVOLVIMENTO DO PROJETO	25
5.7 ESTIMATIVA DE CUSTOS	25
5.8 ORÇAMENTO DO PROJETO	27
5.9 CRONOGRAMA DA ESTRUTURA DE DESENVOLVIMENTO	28
<b>6. LISTA DE EVENTOS</b>	<b>29</b>
<b>7. DIAGRAMAS</b>	<b>30</b>
7.1 DIAGRAMA DE CLASSE	31
7.2 DIAGRAMA ENTIDADE RELACIONAMENTO	32

7.3 DIAGRAMA CASO DE USO GLOBAL .....	32
<b>7.3.1 ESPECIFICAÇÕES DOS CASOS DE USO.....</b>	<b>33</b>
7.3.1.1 CADASTRAR OUVIDOR .....	33
7.3.1.2 ALTERAR OUVIDOR.....	34
7.3.1.3 EXCLUIR OUVIDOR.....	35
7.3.1.4 LISTAR TODOS OS OUVIDORES.....	36
7.3.1.5 LISTAR TODOS OS USUÁRIOS.....	37
7.3.1.6 LISTAR TODAS AS PESSOAS FÍSICA.....	38
7.3.1.7 LISTAR TODAS AS PESSOAS JURÍDICAS .....	39
7.3.1.8 LISTAR TODOS OS ANÔNIMOS.....	40
7.3.1.9 LISTAR TODAS AS MANIFESTAÇÕES .....	41
7.3.1.10 LISTAR MENSAGENS DOS OUVIDORES PARA ALGUMA MANIFESTAÇÃO .....	42
7.3.1.11 LISTAR TODAS AS MENSAGENS DE ALGUM OUVIDOR .....	43
7.3.1.12 RESPONDER MANIFESTAÇÃO.....	44
7.3.1.13 LISTAR TODAS AS MANIFESTAÇÕES .....	45
7.3.1.14 RESPONDER MANIFESTAÇÃO PARA O ADMINISTRADOR .....	46
7.3.1.15 FAZER MANIFESTAÇÃO.....	47
7.3.1.16 ACOMPANHAR MANIFESTAÇÃO .....	48
7.4 DIAGRAMA DE SEQUÊNCIA.....	49
<b>7.4.1 DIAGRAMA DE SEQUÊNCIA ADMINISTRADOR .....</b>	<b>49</b>
<b>7.4.2 DIAGRAMA DE SEQUÊNCIA OUVIDOR.....</b>	<b>50</b>
<b>7.4.3 DIAGRAMA DE SEQUÊNCIA USUÁRIO .....</b>	<b>51</b>
<b>CONCLUSÃO .....</b>	<b>52</b>
<b>REFERÊNCIAS BIBLIOGRAFICAS.....</b>	<b>53</b>
<b>ANEXOS .....</b>	<b>54</b>

## 1. INTRODUÇÃO

Para garantir o direito à informação, orientando o cidadão sobre como obtê-la e como proceder na defesa de sua cidadania, o software será desenvolvido de forma que sua aplicação esteja disponível para a web.

A Ouvidoria é a comunicação direta entre o cidadão e o Poder Público e também é um órgão interlocutor com a sociedade, recebendo dela reclamações, denúncias, sugestões e elogios, estimulando a participação do cidadão no controle e avaliação da prestação dos serviços públicos, favorecendo mudanças e ajustes em suas atividades e processos, em sintonia com as demandas da sociedade, ou seja, um caminho efetivo na busca da qualidade, da transparência e da cidadania.

Aspectos da história da Ouvidoria no Brasil

**1986** - A Prefeitura de Curitiba/PR cria a primeira Ouvidoria pública no País.

**1992** - A Lei nº 8.490/92 cria a Ouvidoria-Geral da República na estrutura regimental básica do Ministério da Justiça.

**1999** - O Estado de São Paulo promulga a lei de proteção ao usuário do serviço público e determina a criação de Ouvidorias em todos os órgãos estaduais.

**2002** - O Decreto nº 4.490/02 cria a Ouvidoria-Geral da República na estrutura regimental básica da Corregedoria-Geral da União.

**2003** - A Lei nº 10.683/03 transforma a Corregedoria-Geral da União em Controladoria-Geral da União, mantendo dentre as suas competências as atividades de Ouvidoria-Geral.

**2004** - A Lei nº 10.689/04 ajusta a denominação de Ouvidoria-Geral da República para Ouvidoria-Geral da União, que, pelo Decreto nº 4.785/03, tem entre outras, a competência de coordenar tecnicamente o segmento de Ouvidorias do Poder Executivo Federal.

## 1.2 OBJETIVO

O sistema a ser implantado na prefeitura tem por objetivo facilitar com que as pessoas possam fazer sugestões, reclamações e elogios, sem precisar se locomover até uma unidade para formalizar suas preocupações, e também buscar a melhoria da qualidade no serviço público prestado na Prefeitura.

## 1.3 JUSTIFICATIVA

O sistema desenvolvido serve para que o cidadão-usuário tenha o direito de manifestar sobre os serviços que a prefeitura presta, assim como sugestões, reclamações ou elogios que serão avaliadas por administradores e ouvidores do website e depois serão publicadas de acordo com as regras e necessidades do órgão público.

## 1.4 PUBLICO ALVO

Todos os cidadãos do município de Assis e Região, o foco do sistema é controlar todas as reclamações, sugestões e elogios e tentar resolver o problema o mais breve possível.

## **2. LEVANTAMENTO DOS REQUISITOS**

### **2.1 FORMA DE LEVANTAMENTO ADOTADA**

O levantamento dos requisitos é umas das partes mais importantes no desenvolvimento do sistema, é neste momento que se deve absorver o máximo de informação do cliente para poder facilitar os processos posteriores que englobam no desenvolvimento do sistema.

### **2.2 RESULTADOS ESPERADOS NA IMPLANTAÇÃO DO SOFTWARE**

Espera-se com a implantação do sistema, a prefeitura, poderá obter maiores informações sobre o município de Assis e tentando melhorar cada vez mais o nosso município.

### **2.3 DETALHAMENTO DO PROBLEMA A SER RESOLVIDO**

Com o grande número de habitantes querendo expor seus pensamentos ou então seus problemas para a cidade de Assis, não tem como atender e retornar rapidamente um parecer, com base dessas carências surge a necessidade de um sistema on-line com vários Ouvidores e Administradores, para poder respondê-los rapidamente.

### **3. ANÁLISE DOS REQUISITOS**

#### **3.1 PROBLEMAS POTENCIAIS**

Cadastrar Ouvidores;

Visualizar todas as manifestações enviadas pelos usuários e mensagens enviadas pelos ouvidores.

#### **3.2 RESTRIÇÕES DO SISTEMA**

Não necessitará de muitos recursos de hardware, porque é uma aplicação desenvolvida para a web, tendo como exigência ter conexão com a internet.

## 4. ESPECIFICAÇÃO DOS REQUISITOS

O sistema desenvolvido controla as manifestações enviadas pelos usuários do município de Assis.

### 4.1 ESCOPO

Atores – Usuário, Ouvidor e Administrador.

Usuário - entrará no website e pode escolher se deseja identificar-se ou ficar como anônimo, então ele enviará uma manifestação, escolhendo-a se a mesma é uma denúncia, elogio ou uma sugestão, feito isso lhe será retornado um número de acompanhamento de sua manifestação.

Ouvidor – Efetuar o login no website, feito isso já lhe será mostrado todas as manifestações dos usuários ordenadas pela data, então assim, ele pode responder as manifestações para que somente o Administrador possa visualiza-las.

Administrador – Efetuar o login no website, será apresentado um menu com varias opções, sendo uma delas, visualizar as manifestações, visualizar as respostas dos ouvidores para tal manifestação e por fim responder manifestação, que está resposta o Usuário irá visualizar quando ele for acompanha-la

## 5. MÉTODO DESENVOLVIMENTO

O sistema foi modelado utilizando-se a metodologia Estruturada.

Para o desenvolvimento do sistema foi utilizada a ferramenta Dreamweaver, com base na linguagem JavaServer Pages.

Para o armazenamento dos dados, será utilizado o banco de dados POSTGRESQL.

### 5.1 AMBIENTES DE DESENVOLVIMENTO

Para o desenvolvimento do sistema foram usadas as seguintes ferramentas.

#### 5.1.1 JAVA

O Java é uma linguagem baseada no C++, seu desenvolvimento teve início em 1991 com a Sun Microsystem em um projeto de codinome Green. O criador da linguagem James Gosling a chamou de Oak. Mas tarde descobriu-se que já existia uma linguagem com esse nome, o nome Java foi sugerido por acaso quando a equipe visitava a cafeteria (Java era o nome de uma cidade de origem de um café importado). Durante algum tempo o projeto passou por dificuldades por causa da falta de mercado, pois o mercado de produtos eletrônicos que era o principal foco da linguagem não estava crescendo como esperado. Porém após a grande popularidade alcançada pela internet em 1993, a Sun viu o grande potencial que o Java tinha para criar páginas mais dinâmicas. A Sun anunciou formalmente o Java em 1995, sendo usada para construir paginas web, sistemas para PC, celulares, palms e todo tipo de dispositivos móveis.

### 5.1.2 DREAMWEAVER

O dreamweaver é um programa gráfico de edição de páginas web. Permite aos Webmasters criar e editar páginas de forma muito similar a de programas de publicação que são usados para a criação de layouts. Com ele você não precisa manipular diretamente o código HTML da página.

Também é um poderoso ambiente de desenvolvimento para a construção de sites dinâmicos que fazem o uso de banco de dados, scripts executados no cliente e tecnologia de servidor, como ASP, JSP, PHP e ColdFusion, e ainda suporta os mais recentes padrões adotados na web, como XHTML, CSS, XML, JAVASCRIPT, XLTS e ActionScript.

É mais do que uma simples ferramenta para o desenvolvimento de aplicativos e layouts, também possui ferramentas que possibilitam o gerenciamento de todos os arquivos relacionados a um site, permitindo o envio destes arquivos para um servidor de web remoto e sua sincronização com os arquivos locais.

### 5.1.3 JAVASERVER PAGES (JSP)

JavaServer Pages (JSP) é uma tecnologia que permite o desenvolvimento rápido de aplicativos baseados na Web que são independentes de plataforma. A tecnologia JSP separa a interface do usuário de geração de conteúdo, permitindo aos designers alterar o layout geral da página sem alterar o conteúdo dinâmico subjacente. JSP também é uma extensão da tecnologia Java Servlet, que são independentes de plataforma, do lado dos servidores módulos que se encaixam perfeitamente em um quadro de servidores Web e pode ser usada para estender as capacidades de um servidor Web com sobrecarga mínima, manutenção e suporte. Ao contrário de outras linguagens de script, servlets não envolvem considerações específicas da plataforma ou modificações, que são componentes da aplicação que são transferidos, a pedido, para a parte do sistema que precisa deles. Juntas, a tecnologia JSP e servlets fornece uma alternativa atraente para outros tipos de dinâmicas Webs scripting / programação, oferecendo: independência de plataforma, melhor desempenho, separação da lógica da exposição, facilidade de administração, extensibilidade na empresa e o mais importante, a facilidade de uso.

## 5.2 FERRAMENTA PARA ANÁLISE

Para poder criar os diagramas do projeto será usado um dos softwares mais conceituados para a criação de diagramas de UML o Astah. Com ele podemos criar diversos tipos de diagramas como, por exemplo:

**Diagrama de Classes** - Os diagramas de classes mostram a estrutura estática do modelo, principalmente os elementos existentes, como classes, sua estrutura interna e seus relacionamentos com outras classes. Eles não mostram informações temporárias.

Um diagrama de classes é apresentado como um conjunto de elementos do modelo declarativos (estáticos) - como classes, pacotes e seus relacionamentos - que são conectados entre si e a seu conteúdo como um gráfico. Os diagramas de classe podem ser organizados em (e pertencentes a) pacotes, mostrando apenas o que é relevante em um determinado pacote.

**Diagrama de Caso de Uso** - descrevem relacionamentos e dependências entre um grupo de *Caso de Uso* e os Atores participantes no processo, é importante observar que Diagramas de Caso de Uso não são adequados para representar o desenho, e não podem descrever os mecanismos internos de um sistema.

Diagramas de Caso de Uso são feitos para facilitar a comunicação com os futuros usuários do sistema, e com o cliente, e são especialmente úteis para determinar os recursos necessários que o sistema deve ter. Diagramas de Caso de Uso dizem *o quê* o sistema deve fazer, mas não fazem — e não podem — especificar *como* isto será conseguido.

**Diagrama de Atividade** - O diagrama de atividades descreve a sequência de atividades, com suporte para comportamento condicional e paralelo. Um diagrama de atividades é uma variante de um diagrama de estados no qual a maioria, se não todos, dos estados é estado de atividade. Portanto, muito da terminologia segue a mesma terminologia de estados.

O diagrama de atividades permite escolher a ordem em que as coisas vão ser feitas. Ele simplesmente determina as regras essenciais de sequências que

devem ser seguidas. Esta é a diferença-chave entre um diagrama de atividades e um fluxograma: os fluxogramas são normalmente limitados a processos sequenciais, enquanto que os diagramas de atividades podem lidar com processos paralelos.

**Diagrama de Pacotes** - Diagramas de pacote, um pacote representa um grupo de classes (ou outros elementos) que se relaciona com outros pacotes através de uma relação de dependência. Um diagrama de pacotes pode ser utilizado em qualquer fase do processo de modelagem e visa organizar os modelos.

O pacote é o elemento básico organizador de um modelo de sistema UML. É possível considerar o sistema todo como um pacote que contém todos os outros pacotes, diagramas e elementos. Um pacote pode conter pacotes subordinados,

diagramas ou elementos únicos e é possível definir a visibilidade de um pacote bem como a visibilidade dos elementos contidos nele.

**Diagrama de Estados** - Um estado é uma condição de um objeto em que ele realiza alguma atividade ou espera um evento. Um objeto pode permanecer em um estado durante um tempo limitado. Um estado tem várias propriedades:

<b>Nome</b>	Uma seqüência de caracteres textuais que distingue o estado de outros estados; um estado também pode ser anônimo, ou seja, não ter nenhum nome.
<b>Ações de entrada/saída</b>	As ações executadas ao entrar no estado ou sair dele.
<b>Transições internas</b>	As transições que são manipuladas sem causar mudança de estado.
<b>Subestados</b>	A estrutura aninhada de um estado, envolvendo subestados separados (ativos seqüencialmente) ou simultâneos (ativos concomitantemente).
<b>Eventos adiados</b>	Uma lista de eventos que não são manipulados no estado, mas são adiados e enfileirados para serem manipulados pelo objeto em outro estado.

Como mostra a Figura, existem dois estados especiais que podem ser definidos para a máquina de estado de um objeto. O estado inicial indica o local

inicial padrão para a máquina de estado ou o subestado. Um estado inicial é representado como um círculo de preenchimento preto. O estado final indica o término da execução da máquina de estado ou que o estado confinado foi concluído. Um estado final é representado como um círculo de preenchimento preto dentro de um círculo sem preenchimento. Os estados final e inicial são, na verdade, pseudo-estados. Nenhum dos dois pode ter as partes usuais de um estado normal, exceto o nome. Uma transição de um estado inicial para um estado final pode ter o complemento completo de características, incluindo uma condição de guarda e uma ação, mas não pode ter um evento trigger.

**Diagrama de Objetos** - O diagrama de objetos é uma variação do diagrama de classes e utiliza quase a mesma notação. O diagrama de objetos é como se fosse o perfil do sistema em certo momento de sua execução.

Existem apenas duas diferenças entre o diagrama de classes os objetos são escritos com seus nomes sublinhados e todas as instâncias num relacionamento são mostradas. Este diagrama é importante para se ter uma visão do sistema em determinado tempo e para se conferir se os dados estão atribuídos corretamente no tempo e se são consistentes.

**Diagrama de Componentes** - diagrama de componentes é apresentado como um conjunto de elementos do modelo declarativos (estáticos) - como componentes, subsistemas e seus relacionamentos - que são conectados entre si como um gráfico. Os diagramas de componentes podem ser organizados em (e pertencentes a) subsistemas de implementação, que mostram apenas o que é relevante em um determinado subsistema de implementação.

A UML não é uma metodologia de desenvolvimento, o que significa que ela não diz para você o que fazer primeiro e nem como projetar o seu sistema, ela lhe auxilia a visualizar seu desenho e a comunicação entre objetos.

Os objetivos da UML são: especificação, documentação, estruturação para sub-visualização e maior visualização lógica do desenvolvimento completo de um sistema de informação.

### 5.3 BANCO DE DADOS

Postgres ou POSTGRESQL é um projeto de Sistema Gerenciador de Banco de Dados open-source que foi iniciado em 1986, na Universidade de Berkeley, na Califórnia.

POSTGRESQL é um sistema de gerenciamento de banco de dados. Ele pode ser qualquer coisa desde uma simples lista de compras a uma galeria de imagens ou a grande quantidade de informação da sua rede corporativa. Para adicionar, acessar, e processar dados armazenados em um banco de dados de um computador, você necessita de um sistema de gerenciamento de banco de dados como o Servidor POSTGRESQL. É um sistema de gerenciamento de bancos de dados relacional, que armazena dados em tabelas separadas em vez de colocar todos os dados em um só local. Isso proporciona velocidade e flexibilidade.

POSTGRESQL é um software Open Source, significa que é possível para qualquer um usar e modificar o programa. Qualquer pessoa pode fazer download do MYSQL pela Internet e usá-lo sem pagar nada. Se você quiser, você pode estudar o código fonte e alterá-lo para adequá-lo às suas necessidades.

O banco de dados POSTGRESQL foi escolhido para o desenvolvimento do projeto pelo motivo dele ser Open Source e também é muito fácil de manusear.

#### 5.3.1 FERRAMENTA PARA GERENCIAR O BANCO DE DADOS

O PGADMIN é uma excelente ferramenta para a administração e desenvolvimento de banco de dados POSTGRESQL, ela permite fazer todas as tarefas necessárias de administração do banco de dados. Esta ferramenta é

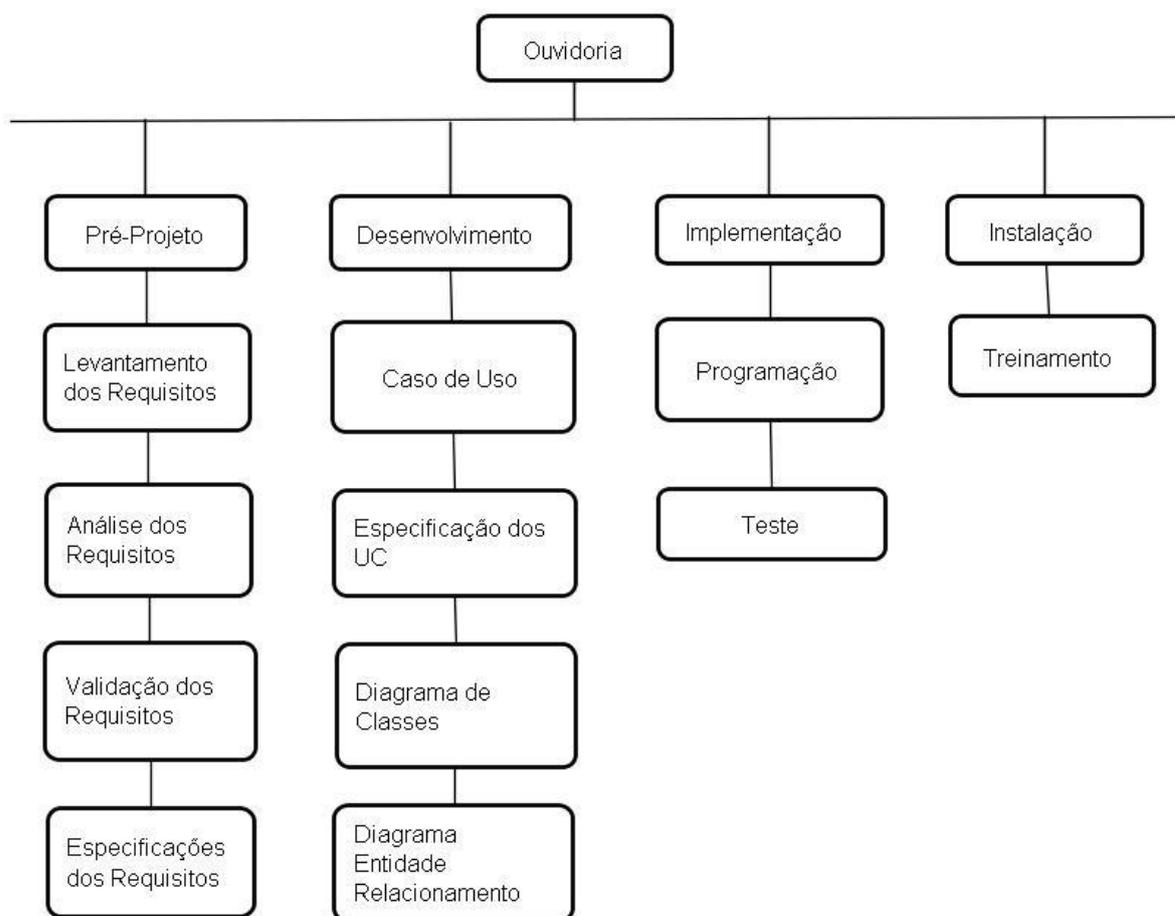
instalada automaticamente quando instalamos o POSTGRESQL, e é caracterizada por ser um aplicativo sofisticado voltado a desenvolvedores profissionais e, ao mesmo tempo de fácil aprendizagem aos iniciantes.

#### 5.4 WBS – WORK BREAKDOWN STRUCTURE

WBS é uma decomposição hierárquica orientada a entrega do trabalho. A WBS deve definir o escopo completo do projeto, ou seja, todas as funcionalidades levantadas junto ao cliente.

Após o projeto e declaração de escopo aprovadas, a WBS deve ser construída, para auxiliar as partes interessadas a visualizar as entregas do projeto.

Dessa forma, a WBS deve ser construída seguindo uma estrutura hierárquica, de forma que as atividades sejam desenhadas partindo da forma genérica à específica, sendo desdobradas a cada nível da mesma.



**Figura 1 Diagrama WBS**

#### 5.4.1 DEFINIÇÕES DAS ATIVIDADES

- ⤴ Levantamento dos requisitos;
- ⤴ Análise dos requisitos;
- ⤴ Validação dos requisitos;
- ⤴ Especificação dos requisitos;
- ⤴ Caso de Uso;
- ⤴ Especificação dos Casos de Uso;
- ⤴ Diagrama de classes;
- ⤴ Diagrama de sequência;
- ⤴ Programação;
- ⤴ Testes;
- ⤴ Treinamento.

## 5.5 SEQUENCIAMENTO DAS ATIVIDADES DEFINIDAS

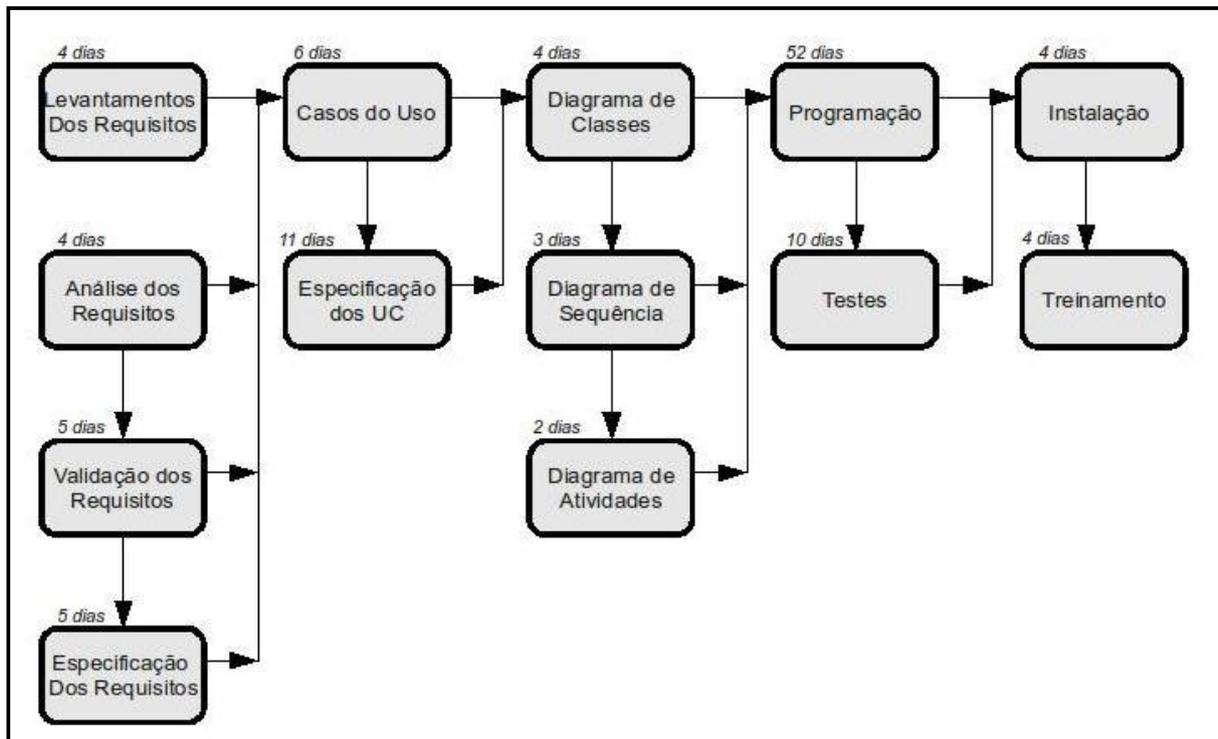


Figura 2 Sequenciamento das Atividades Definidas

## 5.6 RECURSOS NECESSÁRIOS PARA O DESENVOLVIMENTO DO PROJETO

1 (um) Analista de Sistema;

1 (um) Programador Java;

1 (um) Notebook;

1 (uma) Impressora;

Sistema Operacional Windows 7 Ultimate;

IDE para o desenvolvimento: Adobe DreamWeaver;

IDE para modelagem de dados: Astah;

Linguagem para o desenvolvimento: Java;

Servidor de Aplicação: Apache Tomcat;

Bando de dados: POSTGRESQL;

Aplicativos: Microsoft Office 2010.

## 5.7 ESTIMATIVA DE CUSTOS

Analista de Sistema

Quantidade dias: 44

Custo diário: R\$ 80,00 (Oitenta reais)

Custo total: (Total de dias \* Custo diário) = R\$ 3.520,00 (Três mil quinhentos e vinte reais)

Programador

Quantidade dias: 66

Custo diário: R\$ 65,00 (Sessenta e cinco reais)

Custo total: (Total de dias \* Custo diário) = R\$ 4.290,00 (Quatro mil duzentos e noventa reais)

Notebook

Valor Unitário: R\$ 2.200,00 (Dois mil e cem reais)

Dias de uso: 110 dias

Depreciação: (R\$ 2.200,00 / 24 meses) = R\$ 91,67 (Noventa um reais e sessenta e sete centavos) por mês

Custo dia: (R\$ 91,67 / 30 dias) = R\$ 3,05 (Três reais e cinco centavos) por dia

Custo do Notebook: (Total de dias \* R\$ 3,05) = R\$ 335,00 (Trezentos e trinta e cinco reais)

Impressora

Valor Unitário: R\$ 300,00 (Trezentos reais)

Dias de uso: 44 dias

Depreciação: (R\$ 300,00 / 24 meses) = R\$ 12,50 (Doze reais e cinquenta centavos) por mês

Custo dia: (R\$ 12,50 / 30 dias) = R\$ 0,41 (Quarenta e um centavos) por dia

Custo da Impressora: (Total de dias \* R\$ 0,41) = R\$ 18,04 (Dezoito reais e quatro centavos)

Sistema Operacional Windows 7 Ultimate;

Custo: R\$ 499,19

Aplicativos - Microsoft Office 2010.

Custo: R\$ 499,00

Quanto aos outros aplicativos e ferramentas utilizadas no desenvolvimento deste software, ficaram sem custo algum para os clientes. Dentre eles, estão o IDE para o desenvolvimento – **DREAMWEAVER**, IDE para a modelagem de dados – **Astah**, Linguagem para o desenvolvimento: Java, Servidor de Aplicação – **Apache Tomcat** e Banco de dados – **POSTGRESQL**.

## 5.8 ORÇAMENTO DO PROJETO

<b>Recursos</b>	<b>Valor</b>
Analista de Sistema	R\$ 3.520,00
Programador Java	R\$ 4.290,00
Notebook	R\$ 355,00
Impressora	R\$ 18,04
Sistema Operacional Windows 7 Ultimate;	R\$ 499,19
Aplicativos: Microsoft Office 2010.	R\$ 499,00
<b>Custo Total do Projeto:</b>	<b>R\$ 9.181,23</b>

Tabela 01 - Orçamento do Projeto

## 5.9 CRONOGRAMA DA ESTRUTURA DE DESENVOLVIMENTO

Mês	Mar				Abril				Maio				Junho				Julho				Ag.				Set				Out				Nov															
Dia / Semana	0	1	1	2	0	0	1	2	0	1	2	2	0	1	1	2	0	0	1	2	0	1	2	2	0	1	1	2	0	0	1	2	0	0	1	2	0	1	1	2	0	1	1	2				
Lev. Requisitos	x	x																																														
Análise requisito			x	x																																												
Lista de Eventos					x	x																																										
Diagrama UC							x																																									
Descrição UC								x																																								
Diagrama Classe										x																																						
Diagrama Sequência											x																																					
Programação											x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x										
Testes																																															x	x

Tabela 02 – Cronograma das Atividades Definidas

## 6. LISTA DE EVENTOS

Cadastrar Ouvidor

Alterar Ouvidor

Excluir Ouvidor

Listar todos os Ouvidores

Listar todos os Usuários

Listar todas as Pessoas Física

Listar todas as Pessoas Jurídicas

Listar todos os Anônimos

Listar todas as manifestações (Tela - Administrador)

Listar mensagens dos ouvidores para alguma manifestação

Listar todas as mensagens de algum ouvidor

Responder manifestação

Listar todas as manifestações (Tela - Ouvidor)

Responder manifestação para o administrador

Fazer Manifestação

Acompanhar Manifestação

## 7. DIAGRAMAS

### 7.1 DIAGRAMA DE CLASSE

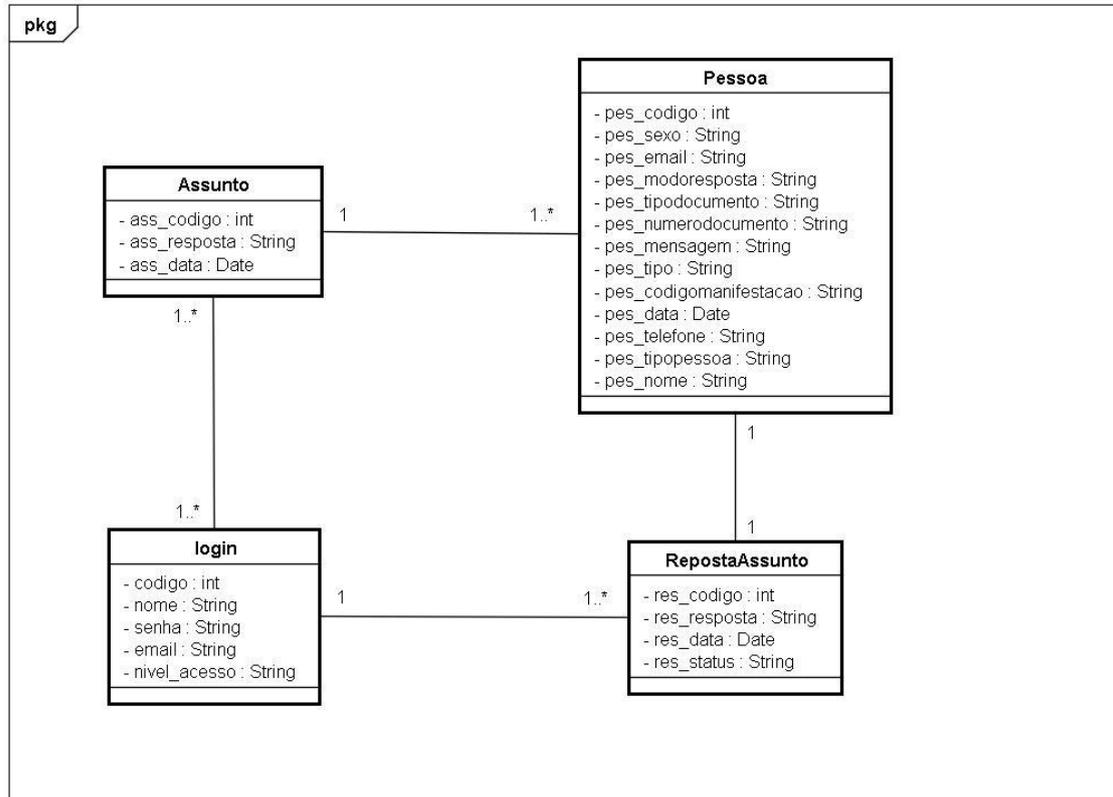
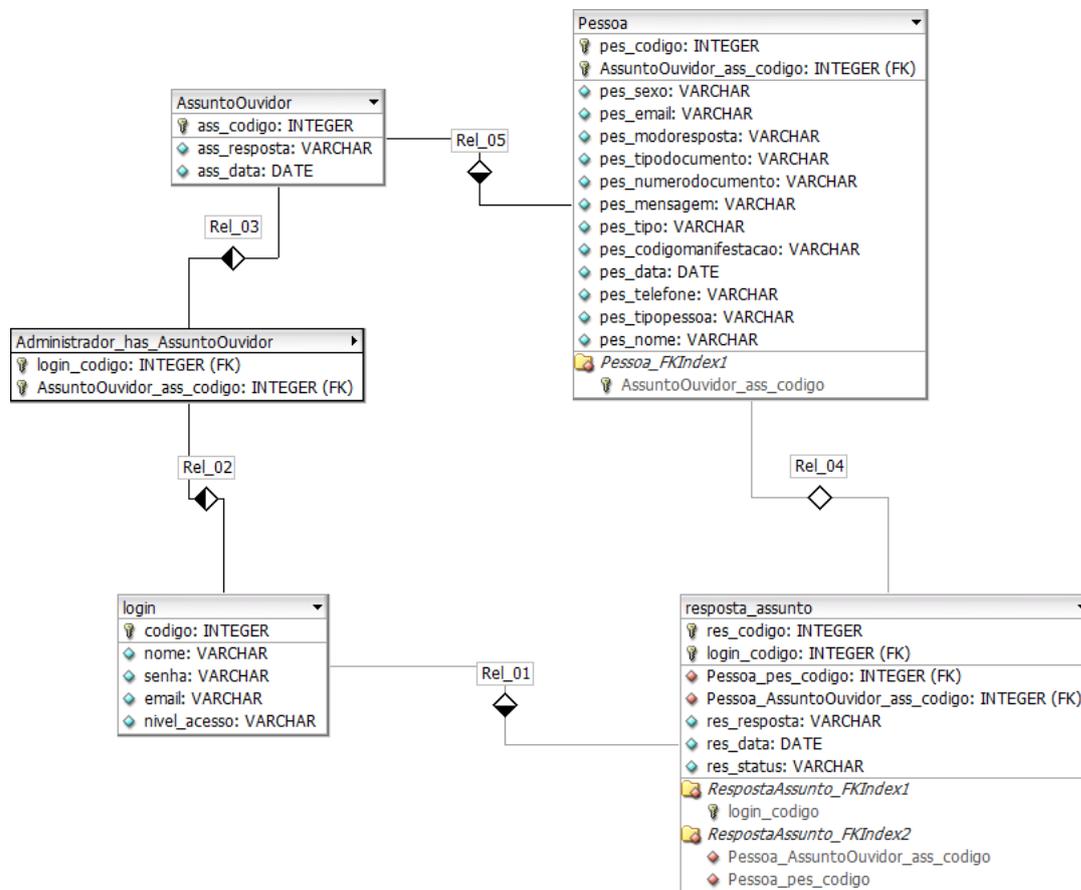


Figura 3 Diagrama de Classe.

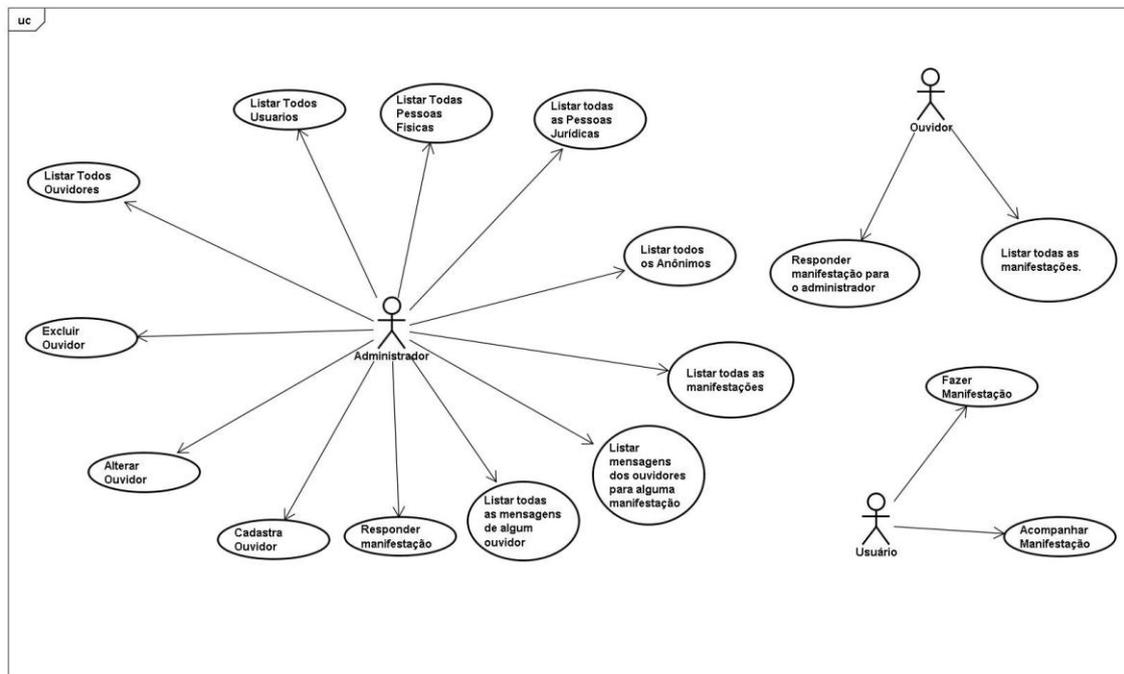
## 7.2 DIAGRAMA ENTIDADE RELACIONAMENTO



**Figura 4 Diagrama De Entidade Relacionamento.**

### 7.3 DIAGRAMA CASO DE USO GLOBAL

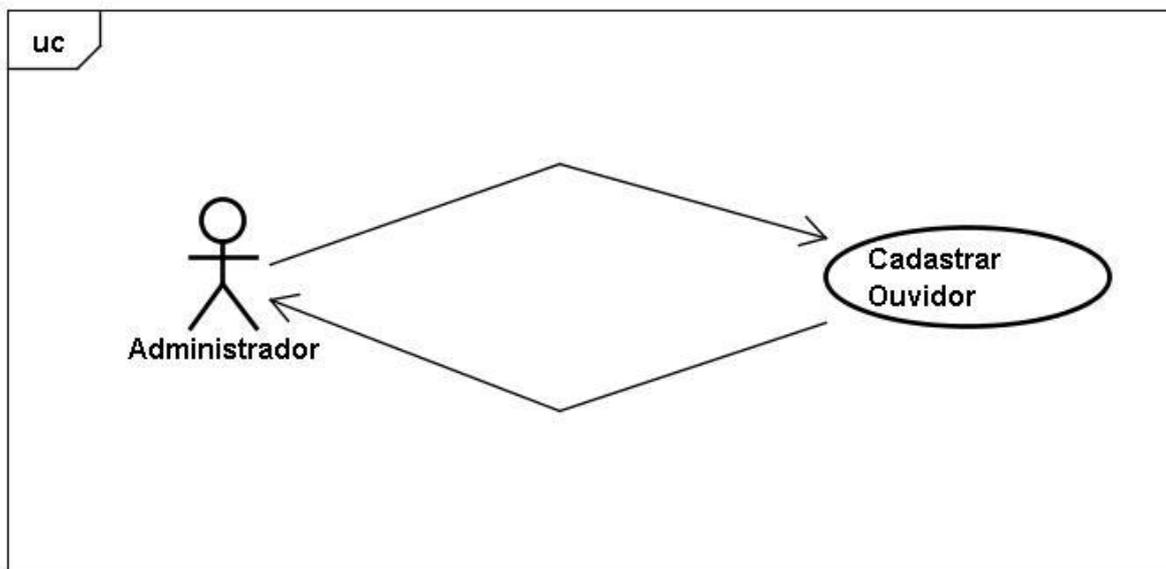
O diagrama de caso de uso global exemplifica as funcionalidades do sistema de forma geral.



**Figura 5 Diagrama Caso de Uso Global.**

### 7.3.1 ESPECIFICAÇÕES DOS CASOS DE USO

#### 7.3.1.1 CADASTRAR OUVIDOR



**Figura 6 Use Case Cadastrar Ouvidor.**

**Pré-condição:**

O Administrador deve ter feito "log-in" e obtido autorização do sistema

**Fluxo de eventos primário:**

O caso de uso começa quando o Administrador seleciona "Cadastrar Ouvidor".

O Administrador passa os dados de log-in e senha para Ouvidor, e escolhe o tipo como Ouvidor.

**Fluxo de eventos secundário:**

A qualquer momento antes de submeter, o Administrador pode selecionar cancelar. O ouvidor não é gravado.

**Pós-condição:**

O ouvidor será gravado, e o sistema retornará a mensagem que o ouvidor foi cadastrado com sucesso.

## 7.3.1.2 ALTERAR OUVIDOR

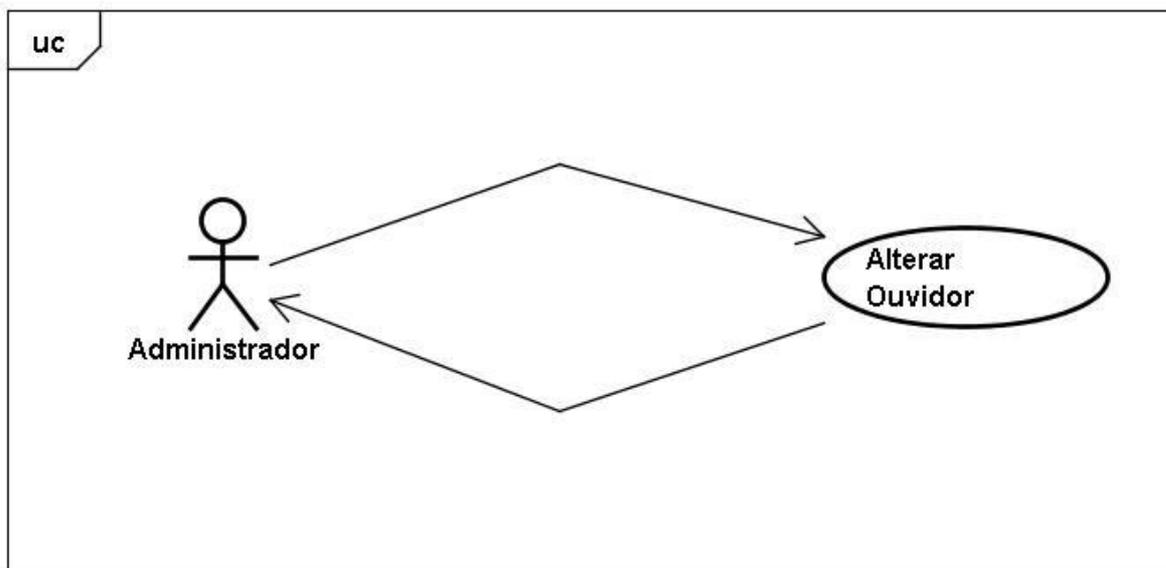


Figura 7 Use Case Alterar Ouvidor.

**Pré-condição:**

O Administrador deve ter feito "log-in" e obtido autorização do sistema

**Fluxo de eventos primário:**

O caso de uso começa quando o Administrador seleciona "Alterar Ouvidor".

O Administrador pode alterar qualquer dado do Ouvidor, exceto o código.

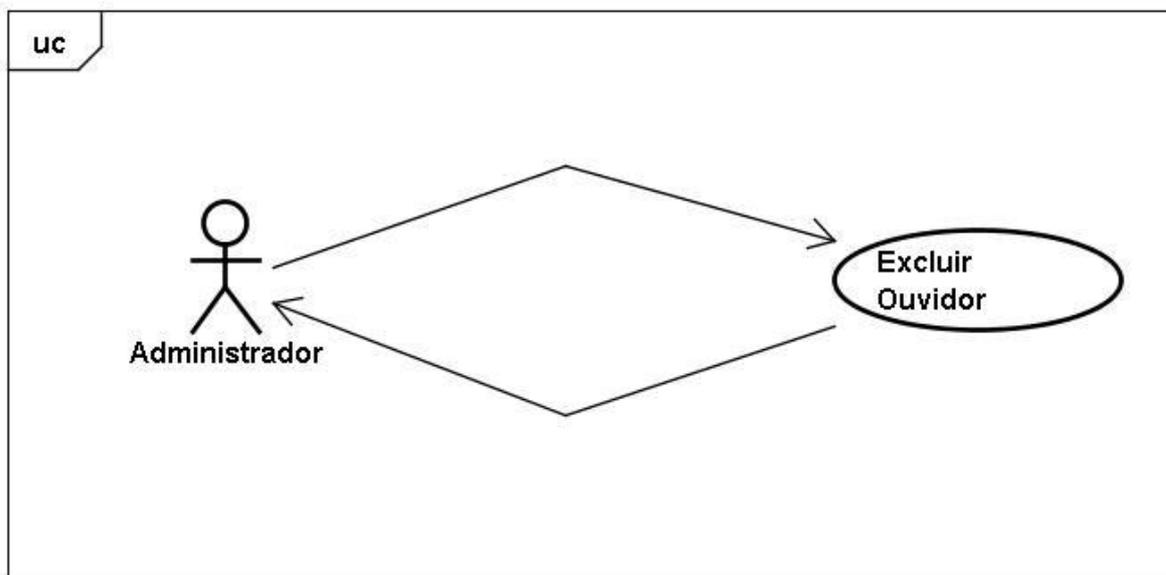
**Fluxo de eventos secundário:**

A qualquer momento antes de submeter, o Administrador pode selecionar cancelar. O ouvidor não é alterado.

**Pós-condição:**

O ouvidor vai ser alterado, e o sistema retornará a mensagem que ele foi alterado com sucesso.

## 7.3.1.3 EXCLUIR OUVIDOR



**Figura 8 Use Case Excluir Ouvidor.**

**Pré-condição:**

O Administrador deve ter feito "log-in" e obtido autorização do sistema

**Fluxo de eventos primário:**

O caso de uso começa quando o Administrador seleciona "Excluir Ouvidor".

O Administrador pode excluir qualquer ouvidor, exceto quando o ouvidor contém mensagens no website.

**Fluxo de eventos secundário:**

A qualquer momento antes de submeter, o Administrador pode selecionar cancelar. O ouvidor não será excluído.

**Pós-condição:**

O ouvidor será excluído, e o sistema retornará a mensagem que ele foi excluído com sucesso.

## 7.3.1.4 LISTAR TODOS OS OUVIDORES

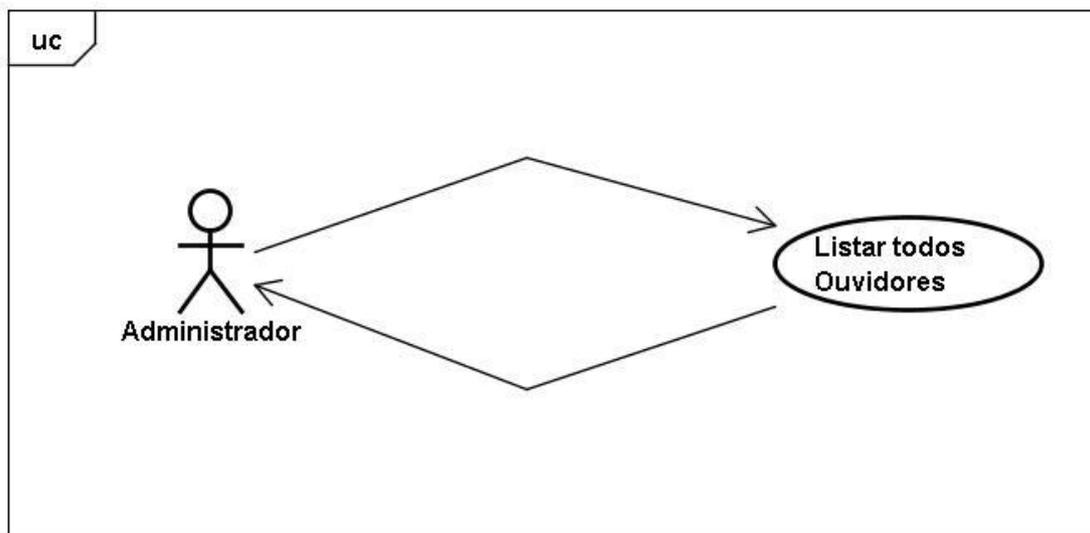


Figura 9 Listar todos os Ouvidores

**Pré-condição:**

O Administrador deve ter feito “log-in” e obtido autorização do sistema.

**Fluxo de eventos primário:**

O caso de uso começa quando o Administrador seleciona “Listar todos Ouvidores” então lhe será gerado todos os Ouvidores cadastrados.

## 7.3.1.5 LISTAR TODOS OS USUÁRIOS

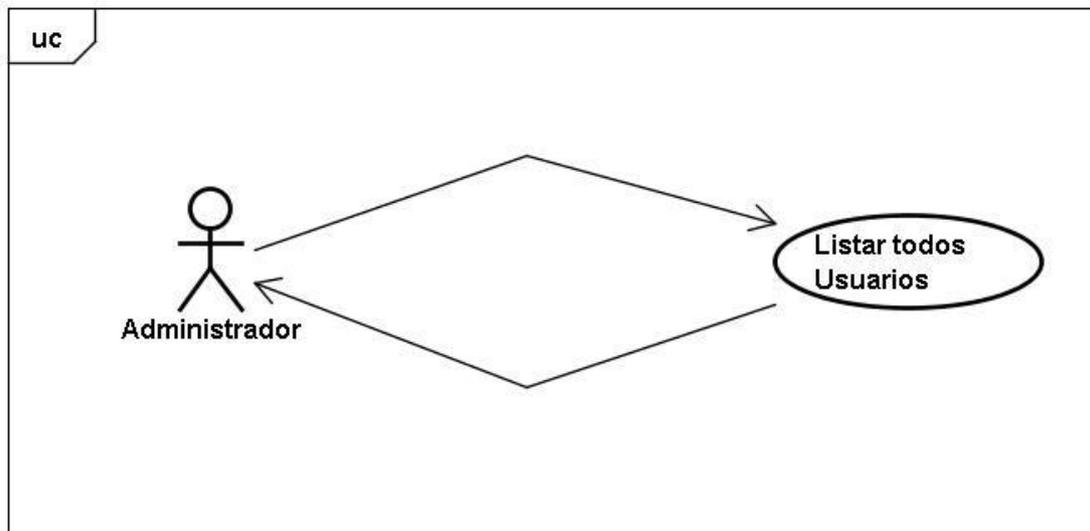


Figura 10 Listar todos os Usuários

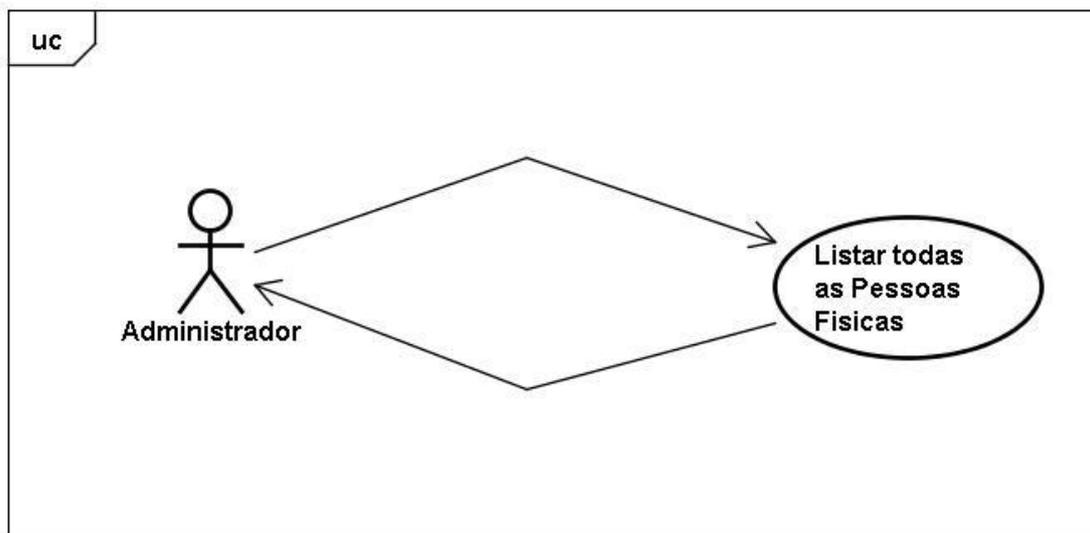
**Pré-condição:**

O Administrador deve ter feito “log-in” e obtido autorização do sistema.

**Fluxo de eventos primário:**

O caso de uso começa quando o Administrador seleciona “Listar todos Usuarios” então lhe será gerado todos os usuários que realizaram manifestações, inclusive os anônimos.

## 7.3.1.6 LISTAR TODAS AS PESSOAS FÍSICA



**Figura 11 Listar todas as Pessoas Físicas**

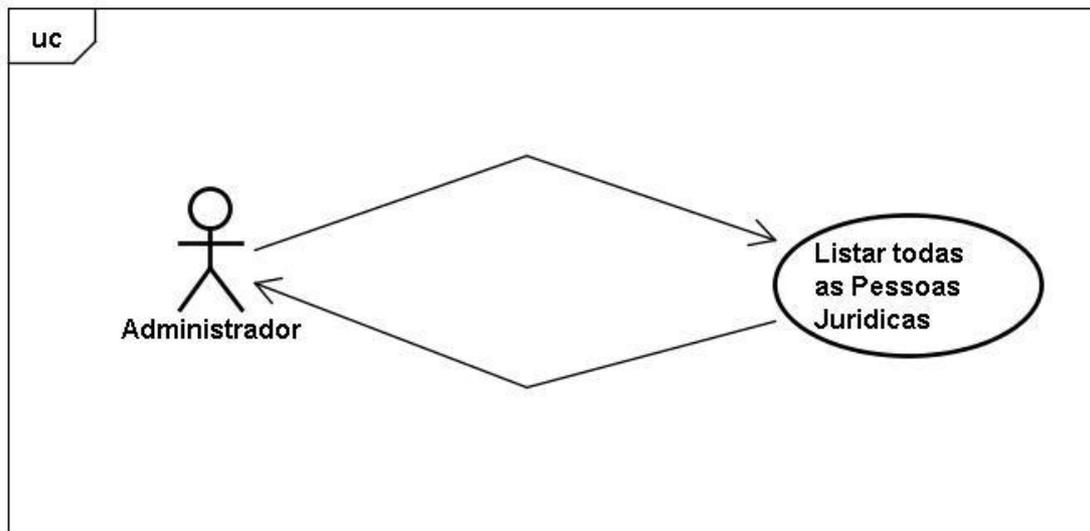
**Pré-condição:**

O Administrador deve ter feito “log-in” e obtido autorização do sistema.

**Fluxo de eventos primário:**

O caso de uso começa quando o Administrador seleciona “Listar todas as Pessoas Físicas” então lhe será gerado todas as Pessoas Físicas que efetuaram alguma manifestação.

## 7.3.1.7 LISTAR TODAS AS PESSOAS JURÍDICAS



**Figura 12 Listar todas as Pessoas Jurídicas**

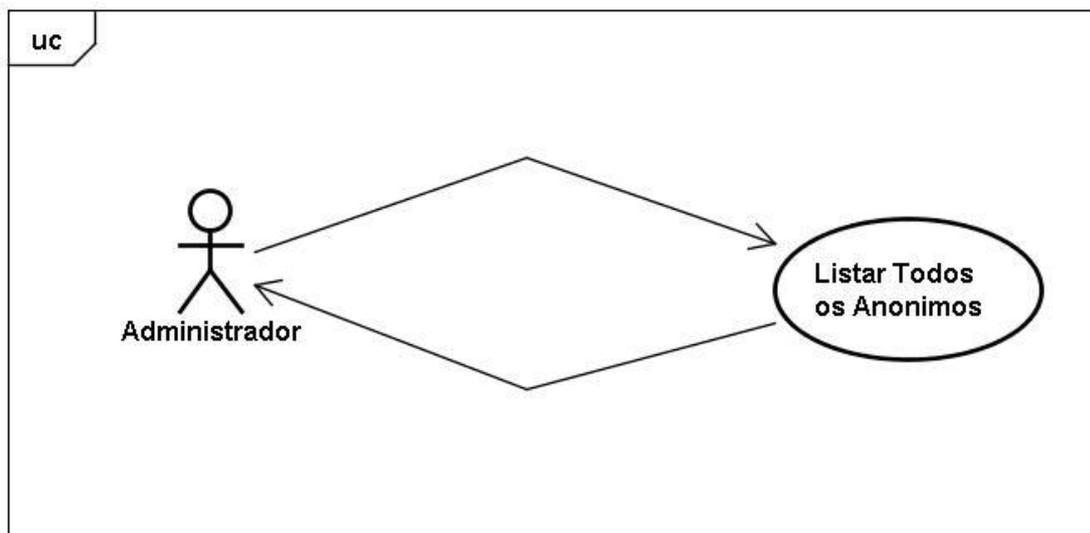
**Pré-condição:**

O Administrador deve ter feito “log-in” e obtido autorização do sistema.

**Fluxo de eventos primário:**

O caso de uso começa quando o Administrador seleciona “Listar todas as Pessoas Jurídicas” então lhe será gerado todas as Pessoas Jurídicas que efetuaram alguma manifestação.

## 7.3.1.8 LISTAR TODOS OS ANÔNIMOS



**Figura 13 Listar Todos os Anônimos**

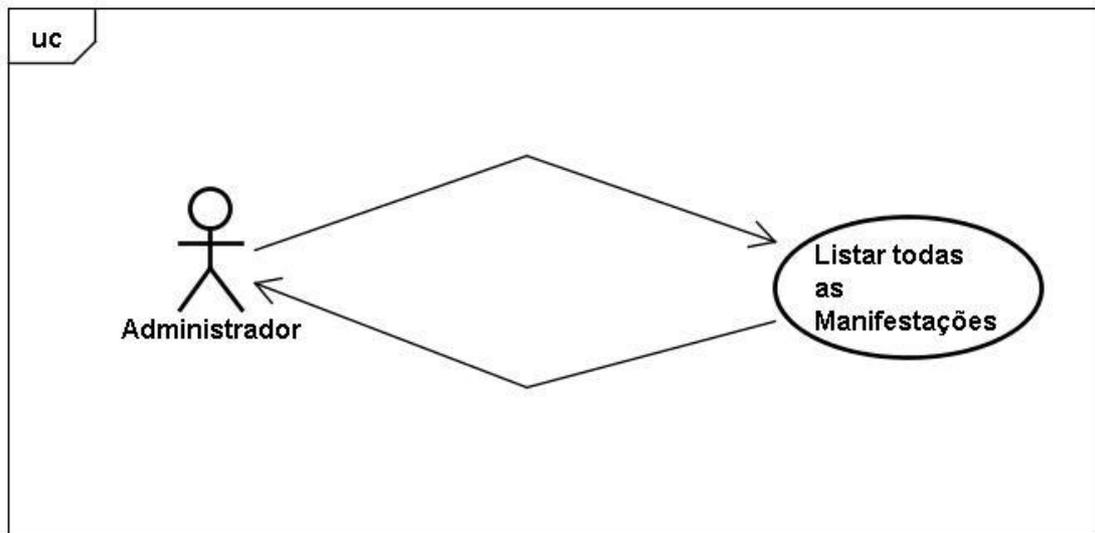
**Pré-condição:**

O Administrador deve ter feito “log-in” e obtido autorização do sistema.

**Fluxo de eventos primário:**

O caso de uso começa quando o Administrador seleciona “Listar todos os Anônimos” então lhe será gerado todos os Anônimos que efetuaram alguma manifestação.

## 7.3.1.9 LISTAR TODAS AS MANIFESTAÇÕES



**Figura 14 Listar todas as Manifestações**

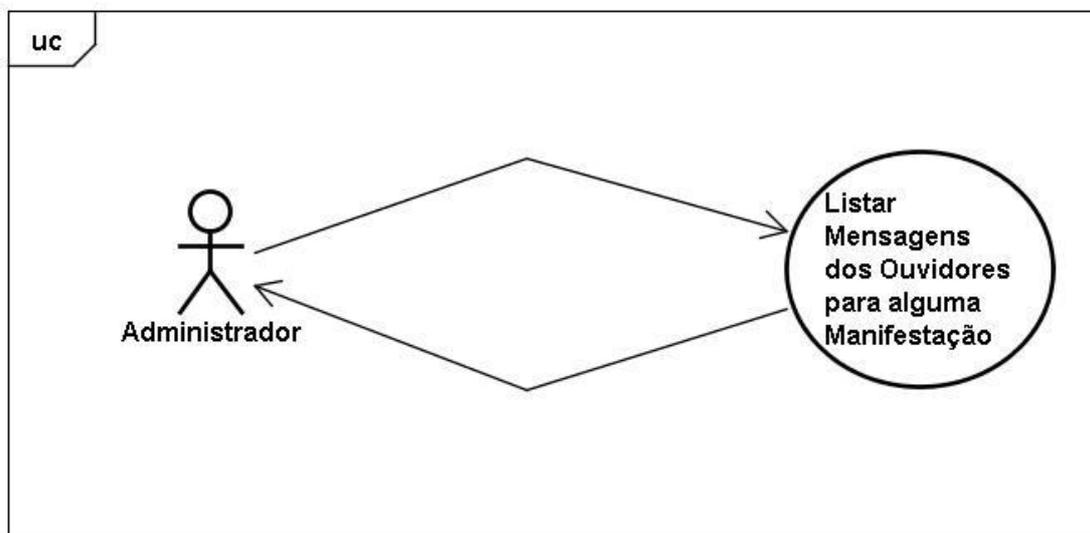
**Pré-condição:**

O Administrador deve ter feito “log-in” e obtido autorização do sistema.

**Fluxo de eventos primário:**

O caso de uso começa quando o Administrador seleciona “Listar todas as Manifestações” então lhe será gerado todas as Manifestações cadastradas.

### 7.3.1.10 LISTAR MENSAGENS DOS OUVIDORES PARA ALGUMA MANIFESTAÇÃO



**Figura 15 Listar Mensagens dos Ouvidores para alguma Manifestação**

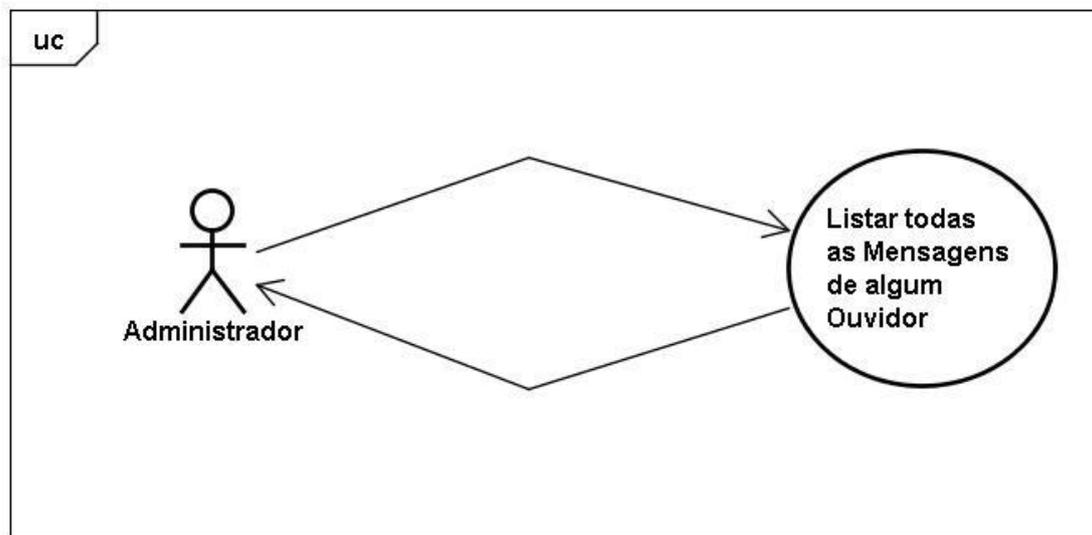
**Pré-condição:**

O Administrador deve ter feito “log-in” e obtido autorização do sistema.

**Fluxo de eventos primário:**

O caso de uso começa quando o Administrador seleciona “Listar Mensagens dos Ouvidores para alguma Manifestação” então lhe serão geradas todas as mensagens dos ouvidores para a manifestação escolhida.

## 7.3.1.11 LISTAR TODAS AS MENSAGENS DE ALGUM OUVIDOR



**Figura 16 Listar todas as Mensagens de algum Ouvidor**

**Pré-condição:**

O Administrador deve ter feito “log-in” e obtido autorização do sistema.

**Fluxo de eventos primário:**

O caso de uso começa quando o Administrador seleciona “Listar todas as Mensagens de algum Ouvidor” então lhe serão geradas todas as mensagens que o ouvidor escolhido enviou.

## 7.3.1.12 RESPONDER MANIFESTAÇÃO

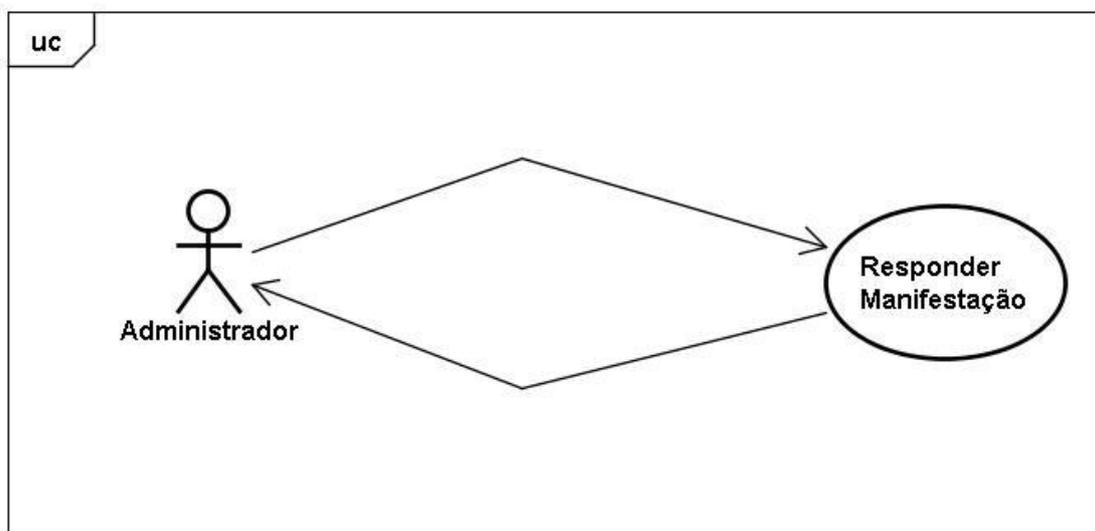


Figura 17 Responder Manifestação

**Pré-condição:**

O Administrador deve ter feito "log-in" e obtido autorização do sistema

**Fluxo de eventos primário:**

O caso de uso começa quando o Administrador seleciona "Responder Manifestação".

O Administrador retornará ao website uma mensagem de resposta para tal sugestão e será publicada para que todos possam ver.

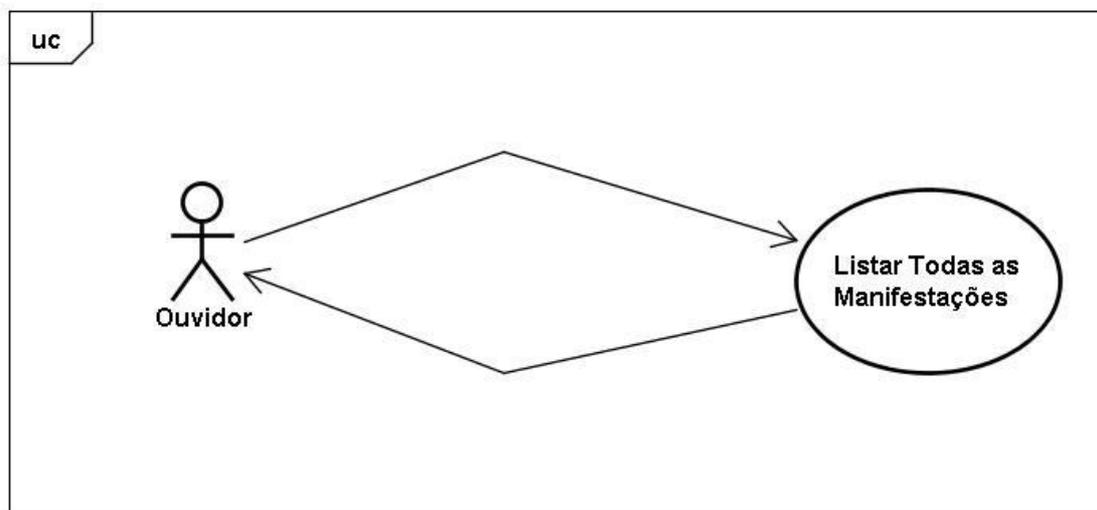
**Fluxo de eventos secundário:**

A qualquer momento antes de submeter, o Administrador pode selecionar cancelar. E a mensagem não será publicada.

**Pós-condição:**

Quando o administrador publicar sua mensagem o sistema retornará uma mensagem que sua publicação no website foi feita com sucesso.

## 7.3.1.13 LISTAR TODAS AS MANIFESTAÇÕES



**Figura 18 Listar Todas as Manifestações**

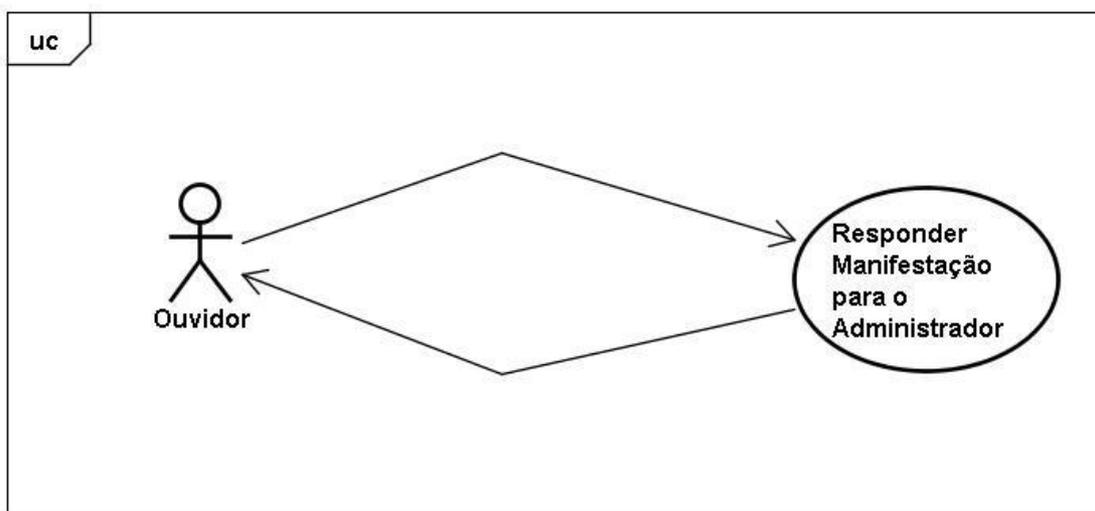
**Pré-condição:**

O Ouvidor deve ter feito “log-in” e obtido autorização do sistema.

**Fluxo de eventos primário:**

O caso de uso começa quando o Ouvidor efetua o login no sistema, ai já lhe será mostrado todas as mensagens dos Usuários.

## 7.3.1.14 RESPONDER MANIFESTAÇÃO PARA O ADMINISTRADOR



**Figura 19 Responder Manifestação para o Administrador**

**Pré-condição:**

O Ouvidor deve ter feito "log-in" e obtido autorização do sistema

**Fluxo de eventos primário:**

O caso de uso começa quando o Ouvidor envia uma resposta a tal manifestação para que somente o Administrador possa visualizar.

**Fluxo de eventos secundário:**

A qualquer momento antes de submeter, o Ouvidor pode selecionar cancelar. E a mensagem não será enviada.

**Pós-condição:**

Quando o Ouvidor enviar sua publicação o sistema retornará uma mensagem que sua publicação foi enviada com sucesso.

## 7.3.1.15 FAZER MANIFESTAÇÃO

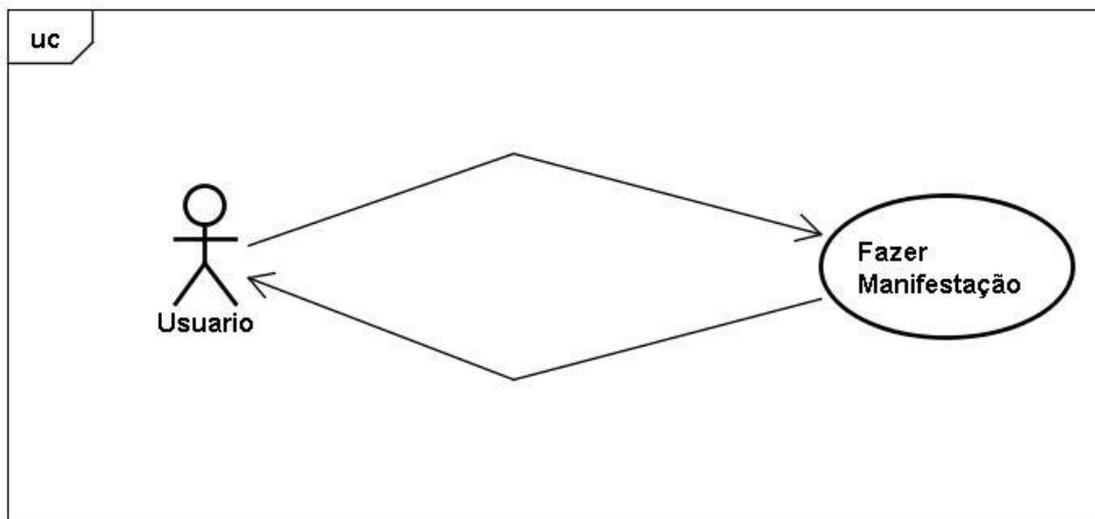


Figura 20 Fazer Manifestação

**Pré-condição:**

O Usuário pode escolher se quer ser identificado ou ser um anônimo.

**Fluxo de eventos primário:**

O caso de uso começa quando o Usuário publica uma sugestão no website para que os Ouvidores possam ver e dar o feedback.

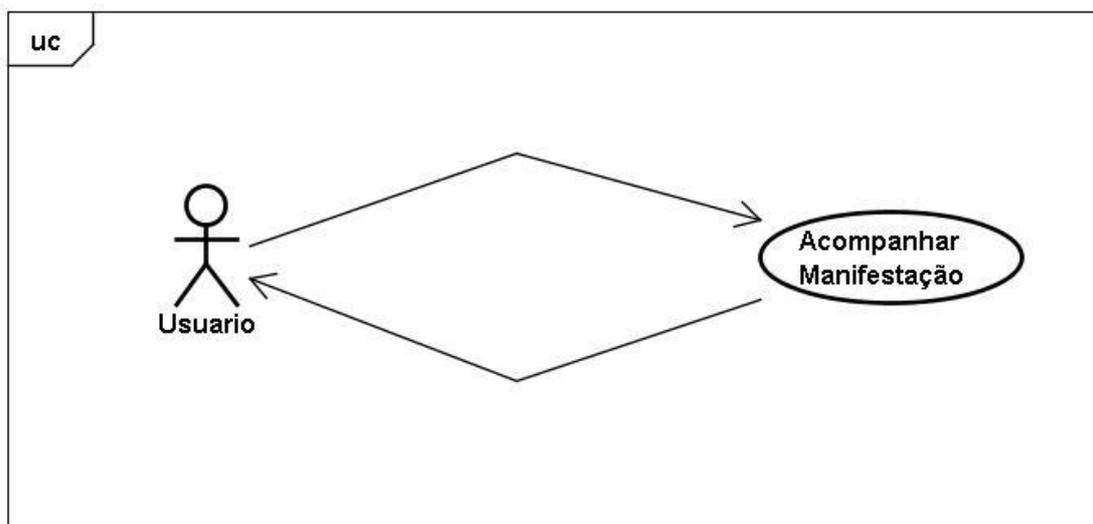
**Fluxo de eventos secundário:**

A qualquer momento antes de submeter, o Usuário pode selecionar cancelar. E a mensagem não será publicada.

**Pós-condição:**

Quando o usuário publicar sua mensagem o sistema retornará uma mensagem que sua publicação no website foi feita com sucesso.

## 7.3.1.16 ACOMPANHAR MANIFESTAÇÃO



**Figura 21 Use Case Acompanhar Manifestação.**

**Pré-condição:**

O Usuário deverá entrar no website.

**Fluxo de eventos primário:**

O caso de uso começa quando Ouvidor seleciona "Acompanhar Manifestação", ele deverá informar o número que lhe foi gerado quando o mesmo enviou a sua manifestação.

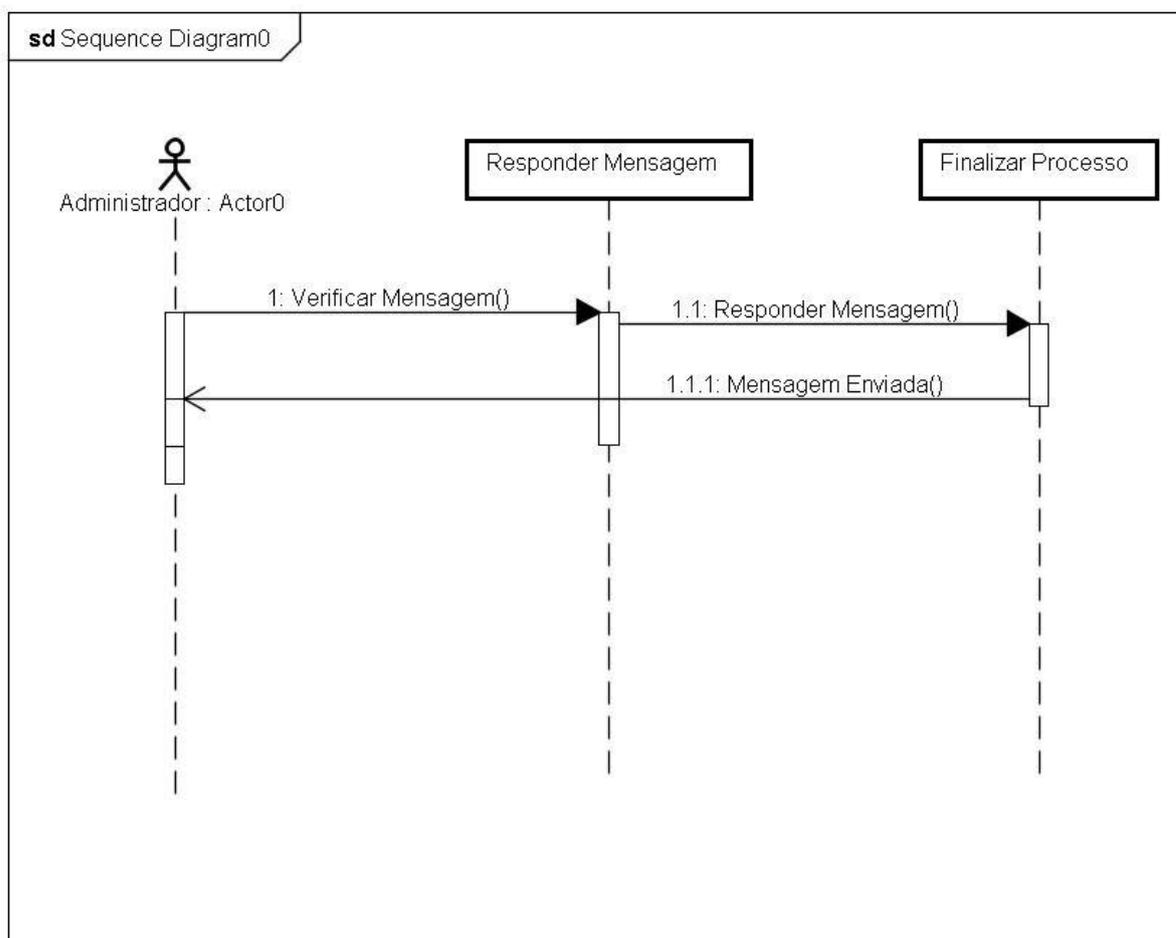
**Fluxo de eventos secundário:**

A qualquer momento antes de submeter, o Ouvidor pode selecionar cancelar. E ele não irá visualizar a sua manifestação.

## 7.4 DIAGRAMA DE SEQUÊNCIA

O diagrama de sequência é muito importante porque ele esclarece os papéis dos objetos em um fluxo, portanto, fornecem entrada básica para determinar responsabilidades de classe e interfaces.

### 7.4.1 DIAGRAMA DE SEQUÊNCIA ADMINISTRADOR



**Figura 22 Diagrama de Sequência Administrador**

## 7.4.2 DIAGRAMA DE SEQUÊNCIA OUIDOR

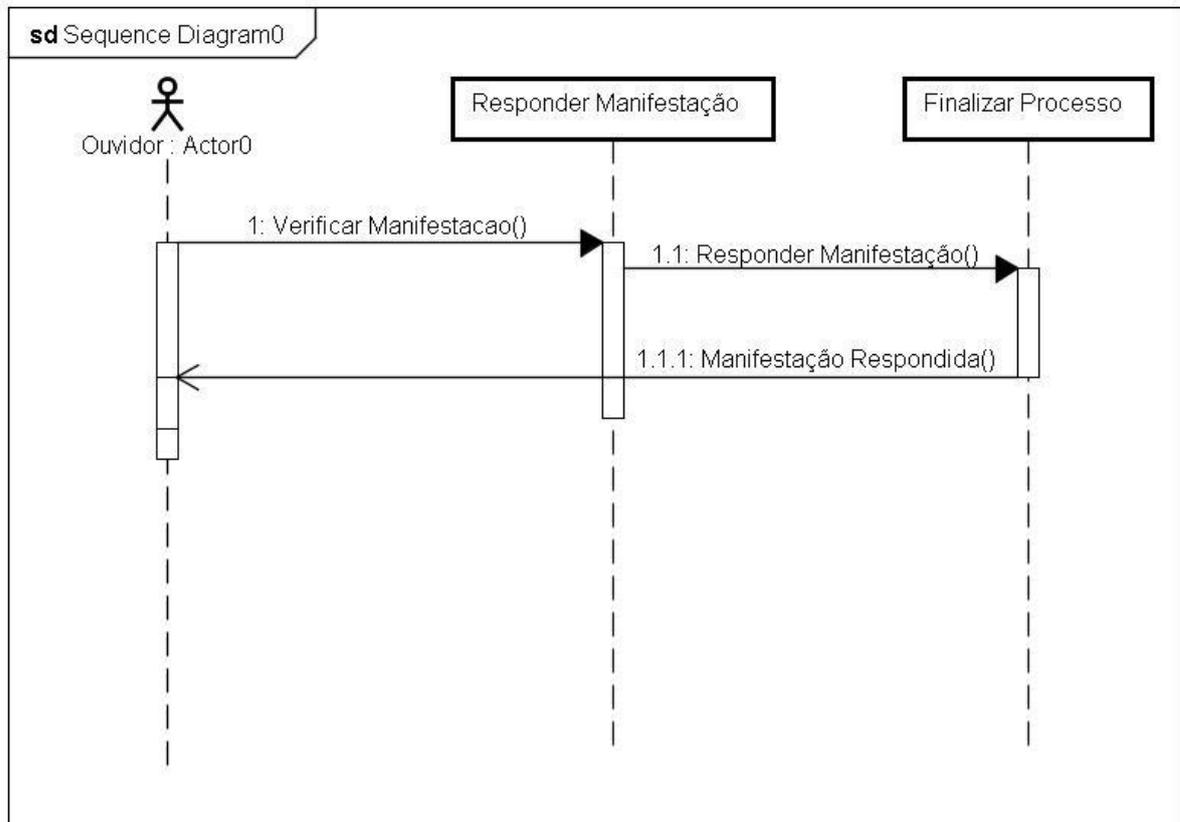


Figura 23 Diagrama de Sequência Ouidor

## 7.4.3 DIAGRAMA DE SEQUÊNCIA USUÁRIO

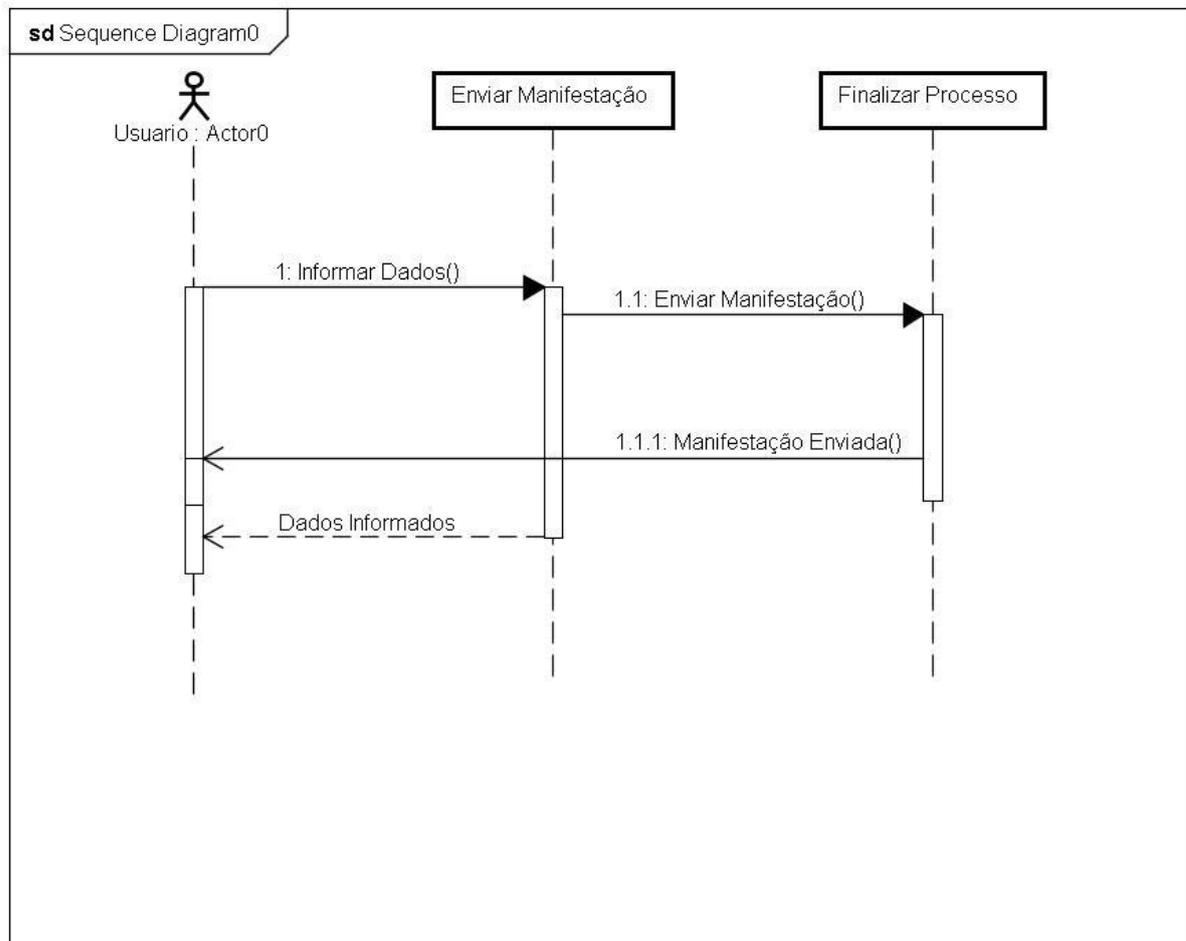


Figura 24 Diagrama de Sequência Usuário

## **CONCLUSÃO**

A implementação do site será de grande ajuda para os moradores do município de Assis-SP, para que possam efetuar suas manifestações (reclamar, sugerir e elogiar) sem nenhuma dificuldade, basta apenas ter acesso á internet, e também não é obrigatório se identificar. O Sistema oferece para seus usuários uma maior praticidade para se manifestar com segurança.

Através do desenvolvimento deste trabalho, podem-se aplicar os conhecimentos adquiridos durante o curso, bem como utilizar boas práticas de programação e ferramentas atuais, em constante disseminação no mercado tecnológico.

## REFERÊNCIAS BIBLIOGRÁFICAS

GONÇALVES, Edson. **Desenvolvendo Aplicações Web com JSP, SERVLETS, JAVASERVER FACES, HIBERNATE, EJB 3 PERSISTENCE E AJAX.** Rio de Janeiro: Editora Ciência Moderna Ltda, 2007.

**Introdução Ouvidoria.** Disponível em:

<http://www.fiocruz.br/cgi/cgilua.exe/sys/start.htm?sid=177>

Welbson de Freitas. **Historia do Java.** Disponível em:

<http://falandodejava.blogspot.com/2007/02/histria-do-java.html>

Samuel Paiva Mucin. **Astah' Community, um software para trabalhar com UMLs.**

Disponível em: <http://www.plantaonerd.com/blog/?p=411>

Paulo Victor Gama Gross de Souza. **WBS** Disponível em:

<http://www.infoq.com/br/articles/user-stories-com-wbs>

**Diagramas.** Disponível em: <http://www.wthree.com>

**PGADMIN.** Disponível em: <http://www.jack.eti.br/www/?p=262>

**DREAMWEAVER.** Disponível em:

<http://www.adobe.com/br/products/dreamweaver.html>

**JAVASERVER PAGES.** Disponível em: -

<http://www.oracle.com/technetwork/java/javaee/jsp/index.html>

**POSTGRESQL.** Disponível em: <http://www.postgresql.org.br/>

## ANEXOS

### TELA INICIAL



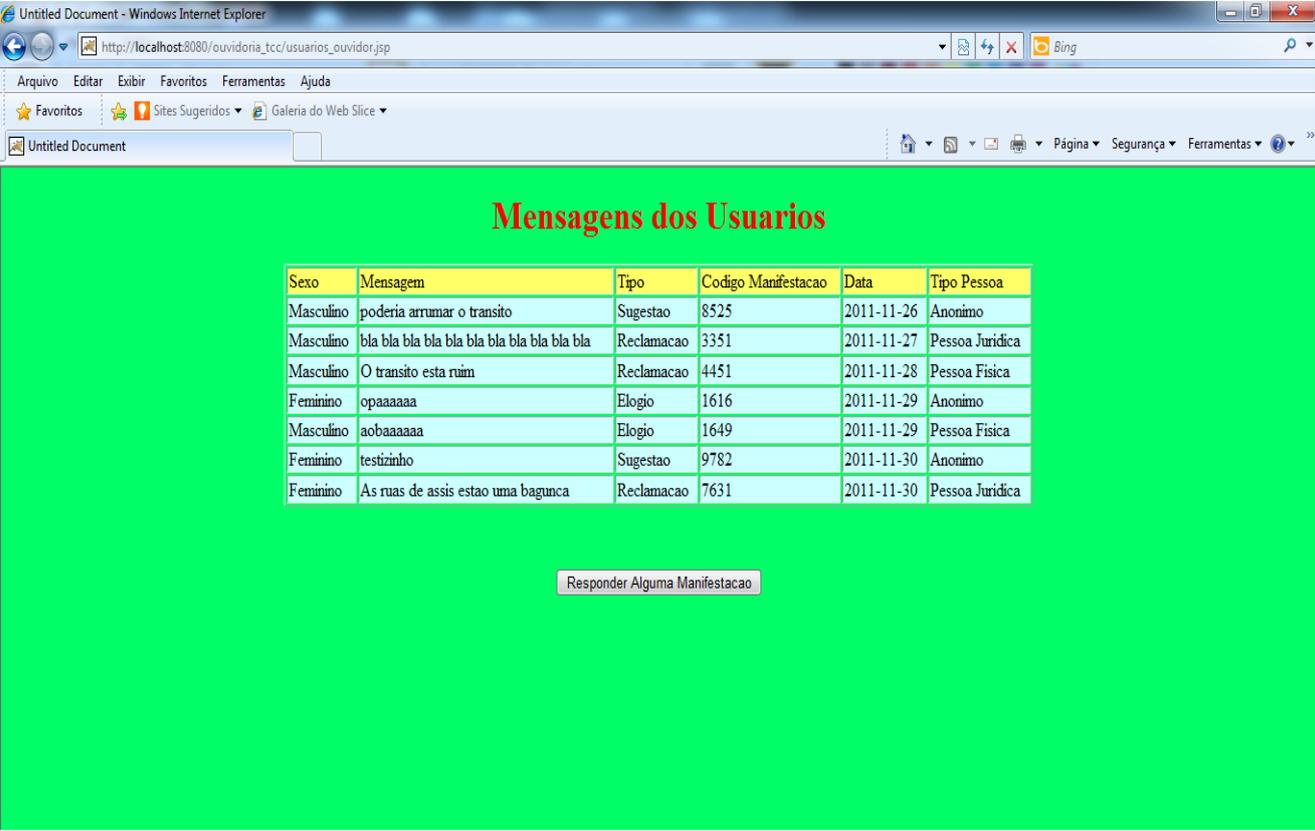
Figura 25 Tela Inicial

## TELA FAZER MANIFESTAÇÃO

The screenshot shows a web browser window with the address bar displaying `http://localhost:8080/ouvidoria_tcc/fazerSolicitacao.jsp`. The browser's menu bar includes 'Arquivo', 'Editar', 'Exibir', 'Favoritos', 'Ferramentas', and 'Ajuda'. The page content features a blue header with the 'Assis' logo and a cityscape. Below the header, the title 'Nova Manifestação - Identificação' is displayed in red, followed by a note: '\* - Campos obrigatórios'. The form contains several fields: 'Nome' (text input), 'Manifestação' (dropdown menu), 'Seco' (dropdown menu), 'Modo Resposta' (dropdown menu), 'Documento de Identificação' (dropdown menu), 'Número Documento' (text input), 'E-Mail' (text input), and 'Telefone' (text input). A 'Manifestação' dropdown menu is located below the 'E-Mail' field. A 'Mensagem\*' text area is positioned below the 'Telefone' field. At the bottom of the form, there are 'Enviar' and 'Limpar' buttons. A footer at the bottom of the page reads: 'Prefeitura Municipal de Assis - Av. Rui Barbosa, 928 - CEP: 19.814-900 - Contato: (18) 3302.3300'. The browser's status bar at the bottom indicates 'Intranet local | Modo Protegido: Desativado' and a zoom level of 50%.

Figura 26 Tela Fazer Manifestação

## TELA OUVIDOR



The screenshot shows a web browser window with the address bar displaying `http://localhost:8080/ouvidoria_tcc/usuarios_ouvidor.jsp`. The page content is on a green background and features the title **Mensagens dos Usuarios** in red. Below the title is a table with the following data:

Sexo	Mensagem	Tipo	Codigo Manifestacao	Data	Tipo Pessoa
Masculino	poderia arrumar o transito	Sugestao	8525	2011-11-26	Anonimo
Masculino	bla bla bla bla bla bla bla bla bla	Reclamacao	3351	2011-11-27	Pessoa Juridica
Masculino	O transito esta ruim	Reclamacao	4451	2011-11-28	Pessoa Fisica
Feminino	opaaaaaa	Elogio	1616	2011-11-29	Anonimo
Masculino	aobaaaaaa	Elogio	1649	2011-11-29	Pessoa Fisica
Feminino	testizinho	Sugestao	9782	2011-11-30	Anonimo
Feminino	As ruas de assis estao uma bagunca	Reclamacao	7631	2011-11-30	Pessoa Juridica

Below the table, there is a button labeled "Responder Alguma Manifestacao". The browser's status bar at the bottom shows "Intranet local | Modo Protegido: Desativado" and a zoom level of 105%.

Figura 27 Tela Ouvidor

## TELA ADMINISTRADOR

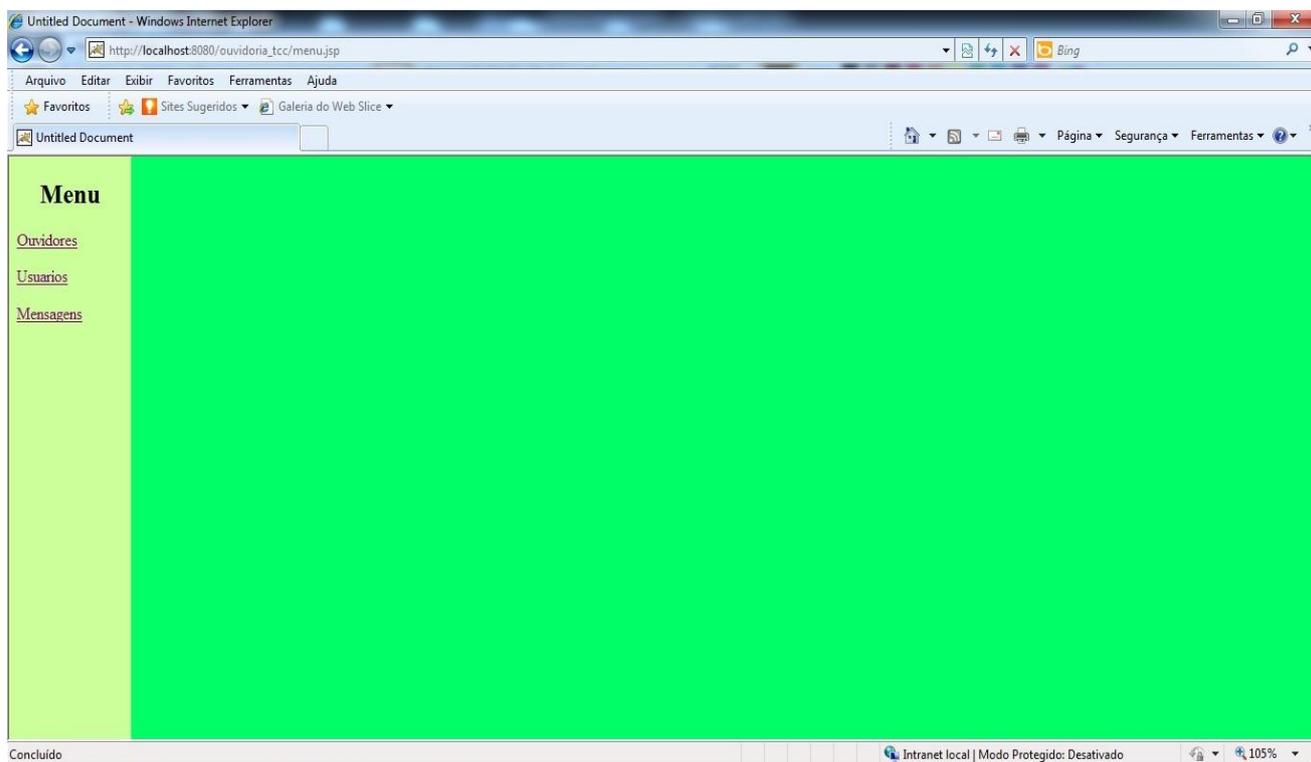


Figura 28 Tela Administrador