



Fundação Educacional do Município de Assis
IMESA - Instituto Municipal de Ensino Superior de Assis

TATIANE SCHWARTZ BORGES

**SISTEMA DE DIAGNÓSTICO MUNICIPAL
INFORMATIZAÇÃO E MANUTENÇÃO DESTES DADOS**

ASSIS

2014

SISTEMA DE DIAGNÓSTICO MUNICIPAL INFORMATIZAÇÃO E MANUTENÇÃO DESTES DADOS

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

Orientador: Drº Almir Rogério Camolesi

Área de Concentração: Desenvolvimento de Sistemas

ASSIS

2014

FICHA CATALOGRÁFICA

SCHWARTZ, Tatiane Borges

Sistema de diagnóstico municipal, informatização e manutenção destes dados/

Tatiane Schwartz Borges. Fundação Educacional do Município de Assis, 2014.

96 p.

Orientador: Dr. Almir Rogério Camolesi

Trabalho de Conclusão de Curso

Instituto Municipal de Ensino Superior de Assis – IMESA.

1. Diagnóstico municipal, 2.Linguagem de programação JAVA,
3.UML.

CDD: 001.61

Biblioteca da FEMA

SISTEMA DE DIAGNÓSTICO MUNICIPAL INFORMATIZAÇÃO E MANUTENÇÃO DESTES DADOS

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: Drº Almir Rogério Camolesi

Analisador: Me. Fábio Eder Cardoso

ASSIS

2014

DEDICATÓRIA

Dedico este trabalho à minha família, e a todas as pessoas que acreditaram em mim e no meu trabalho, apoiando-me para que eu pudesse realizá-lo.

AGRADECIMENTOS

Primeiramente agradeço a Deus por me dar sabedoria e por me fortalecer nas horas de fraquezas. Dele que vem tudo que sou, tudo o que tenho e tudo o que espero em minha vida.

Agradeço também aos meus familiares, Maria Glória Correia Schwartz, Sidnei Pereira Rosa Borges e Talita Schwartz Borges, que me apoiaram durante toda essa caminhada, me dando forças e me motivando, para que eu possa ter um futuro melhor.

Aos meus amigos de sala, Willian Cardoso Moraes, Barbara Macri Camargo, Franciele de Oliveira Dias e principalmente ao meu melhor amigo e namorado, Marcio Almeida de Paula.

Aos meus colegas de trabalho, Gustavo Lazaro e Adauto Matuzaki, que contribuíram para que este trabalho fosse realizado.

Ao projeto Rede Ciranda, pela confiança e oportunidade de desenvolver o sistema proposto. Sou grata ao projeto pelos conhecimentos adquiridos durante todo o período de estágio.

E por fim agradeço ao meu orientador Drº Almir Rogério Camolesi, pela orientação e pelo estímulo durante todo o curso.

“Cada sonho que você deixa pra trás, é um pedaço do seu futuro que deixa de existir.”

Steve Jobs

RESUMO

Neste trabalho será apresentada a análise e implementação de um sistema para a informatização dos dados coletados para a elaboração do diagnóstico municipal da criança e do adolescente. O diagnóstico é uma ferramenta que possibilita a elaboração de estratégias e planos para superar um problema, inovar, criar processos e construir um projeto de município desejado pela sua população.

O sistema tem por objetivo cadastrar as informações do diagnóstico. Ele servirá então para armazenar tais dados e facilitar a realização do diagnóstico pelo conselho municipal da criança e do adolescente (CMDCA).

Com a implementação do sistema o usuário poderá cadastrar informações referentes ao diagnóstico municipal, são elas: informações de atendimentos sociais, saúde, educação, violações de direitos, entre outras, e gerar relatórios dos cadastros efetuados.

Para o desenvolvimento desse projeto foram utilizadas tais tecnologias, como a linguagem de programação JAVA, banco de dados MySQL, *framework* de persistência Hibernate, servidor de aplicação Apache Tomcat, e o ambiente de desenvolvimento IDE Eclipse.

ABSTRACT

This Final Project will show the analysis and implementation of a system built for the informatization of collected data to make a children and teenager's town diagnosis, which is a tool that enables the development of strategies and plans to overcome an issue, innovate, create actions and build a city project desired by its population.

The system aims to register the diagnosis information. It will serve to store such data and ease the execution of the diagnosis by the CMDCA.

By implementing the system, the user will be able to register information referring to the town diagnosis, such as social service information, health, education, rights violation and so on, and generate reports of the records made.

For the development of this project such technologies were used, as the JAVA programming language, MySQL database, Hibernate persistence framework, Apache Tomcat application server, and the Eclipse development environment ID.

LISTA DE FIGURAS

Figura 1- Mapa Mental das funcionalidades do sistema	29
Figura 2- Mapa mental visão geral do sistema.....	30
Figura 3-Segurança pública visão geral	33
Figura 4- UC - Manter categoria denunciante	34
Figura 5 - UC - Manter motivo ameaça/violação	35
Figura 6- UC - Manter organização de segurança pública	36
Figura 7- UC - Manter direito fundamental	37
Figura 8- UC - Manter tipo ameaça/violação	38
Figura 9- UC - Manter característica da queixa.....	39
Figura 10-UC - Manter característica da ameaça/violação	40
Figura 11- UC - Manter características da denuncia	41
Figura 12- UC - Atendimento social visão geral	42
Figura 13- UC - Manter programa de atendimento.....	43
Figura 14- UC - Manter projeto social	44
Figura 15- UC - Manter local do atendimento.....	45
Figura 16- UC - Manter tipo de atendimento	46
Figura 17- UC - Manter atividades desenvolvidas.....	47
Figura 18- UC - Manter entidade social.....	48
Figura 19- UC - Manter características do atendimento social.....	49
Figura 20- UC - Setor da saúde visão geral	50
Figura 21- UC - Manter gestação registrada	51
Figura 22- UC - Manter tipo de contágio (vírus)	52
Figura 23- UC - Manter tipo de vírus	53

Figura 24- UC - Manter estabelecimento assistencial de saúde	54
Figura 25- UC - Manter tipo de vício	55
Figura 26- UC - Manter grau de vício	56
Figura 27- UC - Manter quadro de saúde.....	57
Figura 28- UC - Setor da educação visão geral	58
Figura 29- UC - Manter escola	59
Figura 30- UC - Manter tipo de escola	60
Figura 31- UC - Manter serie.....	61
Figura 32- UC - Manter tipo de série	62
Figura 33- UC - Manter características da matrícula.....	63
Figura 34- Diagrama de Atividade - Cadastro das características da denuncia.....	64
Figura 35- Diagrama de Atividade - Cadastro dos dados da educação	65
Figura 36- Diagrama de Atividades - Cadastro das informações de atendimentos sociais	66
Figura 37- Diagrama de Atividades - Cadastro dos dados da saúde	67
Figura 38- Diagrama de sequência - cadastrar característica da denuncia	68
Figura 39- Diagrama de sequência - cadastrar os dados da educação	69
Figura 40- Diagrama de sequência - cadastrar quadro de saúde	70
Figura 41- Diagrama de sequência - cadastrar atendimento social	71
Figura 42- Diagrama de classe	72
Figura 43- Diagrama de classe - Setor de Segurança Pública.....	73
Figura 44- Diagrama de classe - Setor da Educação.....	74
Figura 45- Diagrama de classe - Atendimentos Sociais.....	75
Figura 46- Diagrama de classe - Setor da saúde	76
Figura 47- Diagrama Entidade Relacionamento.....	77

Figura 48- WBS.....	78
Figura 49 - Orçamento - Analista/Programador.....	79
Figura 50 - Orçamento - Equipamentos	79
Figura 51 - Orçamento - Valor Total.....	80
Figura 52- Cronograma de Atividades.....	80
Figura 53 - Organização dos pacotes	81
Figura 54 - Mapeamentos classe modelo1	82
Figura 55 - Mapeamentos classe modelo2	83
Figura 56 - Classe dao - Métodos persist/List	85
Figura 57 - Classe dao - Métodos merge/remove	86
Figura 58 - Classe bean	87
Figura 59 - Tela de controle de acesso	89
Figura 60 - Tela Menu do Administrador	89
Figura 61 - Tela de Pré-cadastro.....	90
Figura 62 - Tela alterar registro	90
Figura 63 - Tela Menu de usuários.....	91
Figura 64 - Tela gerar cadastro	91
Figura 65 - Tela alterar cadastro	92

LISTA DE TABELAS

Tabela 1- Lista de Eventos.....	32
Tabela 2- Manter categoria denunciante	34
Tabela 3- Manter motivo ameaça/violação	35
Tabela 4- Manter organização de segurança pública.....	36
Tabela 5- Manter direito fundamental.....	37
Tabela 6- Manter tipo ameaça/violação	38
Tabela 7- Manter característica da queixa	39
Tabela 8- Manter característica da ameaça/violação	40
Tabela 9- Manter características da denuncia.....	41
Tabela 10- Manter programa de atendimento	43
Tabela 11- Manter projeto social	44
Tabela 12- Manter local do atendimento	45
Tabela 13- Manter tipo de atendimento.....	46
Tabela 14- Manter atividades desenvolvidas	47
Tabela 15- Manter entidade social	48
Tabela 16- Manter características do atendimento social	49
Tabela 17- Manter gestação registrada.....	51
Tabela 18- Manter tipo de contágio (vírus).....	52
Tabela 19- Manter tipo de vírus.....	53
Tabela 20- Manter estabelecimento assistencial de saúde	54
Tabela 21- Manter tipo de vício	55
Tabela 22- Manter grau de vício.....	56
Tabela 23- Manter quadro de saúde	57

Tabela 24- Manter escola.....	59
Tabela 25- Manter tipo de escola.....	60
Tabela 26- Manter serie	61
Tabela 27- Manter tipo de série.....	62
Tabela 28- Manter características da matrícula	63

LISTA DE ABREVIATURAS E SIGLAS

CMDCA - Conselho Municipal dos Direitos da Criança e do Adolescente

ECA - Estatuto da Criança e do Adolescente

UML - Unified Modeling Language

WBS - Work Breakdown Structure

IDE - Integrated Development Environment

JUDE - Java and UML Developer Environment

WWW - World Wide Web

HTML - HyperText Markup Language

XML - eXtensible Markup Language

UC - Use Cases

Sumário

1 – INTRODUÇÃO	18
1.1 OBJETIVOS DO TRABALHO	20
1.2 PÚBLICO ALVO	20
1.3 JUSTIFICATIVA	20
1.4 ESTRUTURA DO TRABALHO	21
2 - TECNOLOGIAS E FERRAMENTAS PARA O DESENVOLVIMENTO DO SISTEMA	22
2.1 LINGUAGEM UNIFICADA DE MODELAGEM (UML)	22
2.1.1 Diagrama de casos de uso (UC)	23
2.1.2 Diagrama de atividades	23
2.1.3 Diagrama de sequência	23
2.1.4 Diagrama de classe	23
2.1.5 Diagrama entidade e relacionamento (DER)	24
2.2 JAVA	24
2.3 JASPERREPORT	25
2.4 IREPORT	25
2.5 JAVA SERVE FACES (JSF)	26
2.6 HIBERNATE	26
2.7 ECLIPSE IDE	26
2.8 MYSQL	27
3 - ANÁLISE, ESPECIFICAÇÃO E PROJETO	28
3.1 ANÁLISE E ESPECIFICAÇÃO	28
3.1.1 Mapa mental	28

3.1.2 Lista de eventos.....	31
3.1.3 Diagrama de casos de uso (UC)	33
3.1.4 Diagramas de atividades.....	64
3.1.5 Diagramas de sequência.....	68
3.1.6 Diagrama de classe	72
3.1.7 Diagrama entidade e relacionamento (DER)	77
3.2 PLANEJAMENTO DO PROJETO	78
3.2.1 Estrutura analítica de trabalho (WBS).....	78
3.2.2 Orçamento	79
3.2.3 Cronograma de atividades	80
4 - DESENVOLVIMENTO DO SISTEMA	81
4.1 ORGANIZAÇÃO DO PROJETO	81
4.1.1 Pacote modelo	82
4.1.2 Pacote dao.....	84
4.1.3 Pacote bean.....	86
4.2 INTERFACES DO SISTEMA	88
5 - CONCLUSÃO	93
REFERÊNCIAS BIBLIOGRÁFICAS	95

1 – INTRODUÇÃO

O Projeto Rede Ciranda da Criança e do Adolescente de Assis nasceu em 2010, de parceria com a Associação Filantrópica Nosso Lar e a Fundação Telefônica. O Projeto faz a integração das áreas de assistência social, educação, saúde, segurança pública e organizações do terceiro setor. (XAVIER, et. al..., 2012, p.15).

Com essa integração, o Rede Ciranda oferece alguns benefícios às entidades associadas, dentre eles, cursos de capacitação e encontros para que seja potencializado a atuação das entidades e projetos sociais, melhoria da atuação do Poder Público, otimização e soluções em realizar o diagnóstico sobre a realidade da criança e do adolescente de Assis que sustenta e apoia o CMDCA na formulação de políticas públicas. (XAVIER, et. al..., 2012, p.15-16).

Em 2009 após parceria firmada entre a Fundação Telefônica e o Conselho Municipal dos Direitos da Criança e do Adolescente (CMDCA)¹, deu-se início ao trabalho de diagnosticar os problemas que atingem a população infanto juvenil no município de Assis, afim de analisar as condições e capacidades disponíveis para enfrentar esses problemas e propor ações que garantam os direitos previstos no Estatuto da Criança e do Adolescente(ECA). (XAVIER, et al..., 2011, p.163).

Com a falta de um instrumento que pudesse orientar os principais passos para realizar ações, corria-se o risco de equívocos de várias ordens. Com a necessidade de ter uma visão dos principais problemas, deu-se início o processo de diagnostico.

¹www.cmda-ssisorg.br

O diagnóstico é uma ferramenta que possibilita a elaboração de estratégias e planos para superar um problema, inovar, criar processos e construir um projeto de Município desejado pela sua população estabelecendo as prioridades de ações integradas e sustentáveis. Ou seja, diagnosticar os problemas que atingem a população infanto juvenil, analisar as condições para enfrentar esses problemas e propor ações.

Segundo Ribas (2011), é importante que os diagnósticos sejam qualificados, pois através desses dados coletados é que se pode ter um resultado significativo, para que se possa tornar realidade os princípios e objetivos estabelecidos no Estatuto da Criança e do Adolescente (ECA).

Além disso, para que haja melhorias das condições de vida das crianças e dos adolescentes, é necessário que seja feito bons diagnósticos, que relatem todos os problemas que afetam a população infanto juvenil no Município.

É de grande importância que todos sejam informados sobre as condições que hoje vive a população infanto juvenil, conhecer a realidade das Crianças e dos Adolescentes é essencial para que se possa conceber e implantar medidas de intervenção consistentes.

Para que isso ocorra foi realizado um estudo sobre os diagnósticos já existentes, e está sendo desenvolvido um sistema para manutenção dos dados manipulados pelo mesmo. A funcionalidade do sistema é extrair informações do diagnóstico municipal, que são armazenadas em papéis e arquivos eletrônicos, são elas: informações de medidas protetivas e socioeducativas, defesa de direitos e as informações sócias básicas (educação, saúde, cultura, esporte, lazer e assistência à família).

Com essas informações inseridas no sistema de diagnóstico e armazenadas em banco de dados, pode-se então gerar relatórios e fazer comparações ano a ano. Com isso as entidades sociais terão visão dos atendimentos e da real situação da população infanto juvenil, poderão diagnosticar os problemas que afetam a população, e por fim compartilhar informações. Além disso, o Poder Público e o CMDCA poderão traçar metas e planejar ações, para um futuro melhor.

1.1 OBJETIVOS DO TRABALHO

O Sistema de Diagnóstico proposto tem por objetivo extrair dados do diagnóstico municipal e promover uma análise compartilhada da situação da Criança e do Adolescente do município de Assis. Por meio dos dados que foram coletados para o diagnóstico, a ideia é informatizar a manutenção dos mesmos e a geração de relatórios.

1.2 PÚBLICO ALVO

O sistema serviu para as organizações como os Conselhos (CMDCA), Poderes Públicos e Instituições sociais que atendem Crianças e Adolescentes no município.

1.3 JUSTIFICATIVA

A informatização dos dados coletados referentes à situação da Criança e do Adolescente no município foi de extrema importância, pois ficou mais fácil de concretizar os princípios e objetivos propostos para a melhoria de vida da população infanto juvenil.

As informações para a realização do diagnóstico são coletadas por meio de questionários que são aplicados nas organizações e registradas em papéis e arquivos eletrônicos, sendo assim as organizações não conseguem ter uma fácil visualização dos resultados do diagnóstico.

O sistema serviu então para armazenar tais dados e facilitar a realização do diagnóstico pelo CMDCA, por meio da geração destes relatórios. As organizações poderão armazenar suas informações em banco de dados e não mais em papéis. Assim, terão mais facilidades para analisar as condições e soluções para esses problemas e propor ações.

1.4 ESTRUTURA DO TRABALHO

O trabalho será dividido em 5 capítulos, sendo eles:

- **Capítulo 1 - Introdução:** O primeiro capítulo apresenta a contextualização e a justificativa para o desenvolvimento da proposta do trabalho.
- **Capítulo 2 - Tecnologias e Ferramentas para o desenvolvimento do sistema:** Neste capítulo serão apresentadas as tecnologias e ferramentas utilizadas no desenvolvimento do Sistema de Diagnóstico.
- **Capítulo 3 - Análise, especificação e projeto:** Este capítulo apresenta informações das ferramentas utilizadas para a modelagem e análise do sistema proposto e o planejamento do projeto.
- **Capítulo 4 - Desenvolvimento do sistema:** Neste capítulo será exibido um detalhamento de como esta sendo o desenvolvimento da aplicação, a organização dos pacotes, a distribuição das camadas e a interface criada para interagir com o usuário final.
- **Capítulo 5 - Conclusão e Trabalhos futuros:** Por fim este capítulo apresentará a conclusão e os trabalhos futuros.
- **Referências Bibliográficas.**

2 - TECNOLOGIAS E FERRAMENTAS PARA O DESENVOLVIMENTO DO SISTEMA

Para a Análise Orientada a Objetos, foram utilizados os conceitos do padrão de Linguagem de Modelagem Unificada (UML) e a ferramenta de modelagem utilizada foi o Astah² *Professional*.

A implementação foi realizada com base na tecnologia Java³, pois é uma linguagem de alto nível orientada a objetos. O ambiente de desenvolvimento utilizado foi o IDE Eclipse. Como base de dados, foi utilizado o banco de dados MySQL⁴, o qual responde as instruções de seleção e manipulação de dados.

2.1 LINGUAGEM UNIFICADA DE MODELAGEM (UML)

A Linguagem Unificada de Modelagem (UML) é uma linguagem para descrever graficamente “*Software*”. O objetivo da UML é proporcionar aos desenvolvedores um conjunto de ferramentas para análise e modelagem. A UML é utilizada no apoio às fases de análise, projeto e implementação, pois nos permite pensar antes de codificar. Trabalhar com diagramas nos ajuda ter uma visão clara das funcionalidades do sistema. Ajuda a apontar e/ou prever inconsistências e omissões. Por fim, possibilita documentar os artefatos do sistema. (GOÉS, 2014, p.16-17)

² <http://www.astah.net/>

³ <https://www.java.com>

⁴ <http://www.mysql.com/>

2.1.1 Diagrama de casos de uso (UC)

A principal função do diagrama de casos de uso (UC) é apresentar quais são os módulos que compõe o sistema, quais são os usuários e quais funções cada um vai desempenhar para o funcionamento do sistema. (GÓES, 2014).

2.1.2 Diagrama de atividades

O diagrama de atividade é definido pela UML para representação do fluxo de controle de uma determinada atividade para a outra, de forma que possa ser ilustrada uma modelagem dos aspectos dinâmicos do sistema. (GÓES, 2014).

2.1.3 Diagrama de sequência

O diagrama de sequência descreve a sequência de comunicação entre os objetos de um sistema de informação. Seus principais objetivos são: documentar casos de uso, mostrar como os objetos do sistema se comunicam validar se todas as operações das classes foram identificadas e declaradas ou ainda validar a existência de um objeto necessário ao funcionamento do sistema. (GOÉS, 2014).

2.1.4 Diagrama de classe

O diagrama de classe ilustra um conjunto de classes e seus relacionamentos de forma estática de um sistema, eles são importantes não só para documentação, mas também para o desenvolvimento do sistema contribuindo de forma impactante para a consolidação do escopo das funcionalidades do projeto. (GÓES, 2014).

2.1.5 Diagrama entidade e relacionamento (DER)

O Diagrama Entidade e Relacionamento - DER é o diagrama de documentação do banco de dados relacionais, sendo responsável por mostrar de maneira gráfica os relacionamentos entre as entidades do banco de dados. (HEUSER, 2009).

2.2 JAVA

A Sun Microsystems financiou um projeto de pesquisa corporativa interna (**Projeto Green**) em 1991. Esse projeto resultou em uma linguagem de programação baseada na linguagem C++. O seu criador James Gosling, chefe do projeto, chamou essa linguagem de Oak em homenagem a uma árvore de carvalho vista por sua janela na Sun. (DEITEL, 2010, p. 6).

Então, mais tarde foi descoberto que já havia uma linguagem de computador com esse nome. Quando a equipe da Sun visitou uma cafeteria local, foi sugerido o nome Java, pois era o nome da terra de origem do café, que os programadores da equipe apreciavam nessa cafeteria, por isso que a logo do Java é um café. (DEITEL, 2010, p. 6).

O projeto de pesquisa passou por algumas dificuldades. O mercado para dispositivos eletrônicos inteligentes destinados ao consumidor final não estava se desenvolvendo tão rapidamente como a Sun imaginou. Em 1993 a Web explodiu em popularidade, foi então que a Sun viu o potencial de utilizar Java para adicionar conteúdo dinâmico, como interatividade e animações, às páginas da Web. Foi isso que deu nova vida ao projeto. (DEITEL, 2010, p. 6).

Em maio de 1995 a Sun anunciou o Java em uma conferência, o Java chamou a atenção da comunidade de negócios por causa do enorme interesse na Web. Agora o Java é bastante utilizado para desenvolver páginas Web, desenvolver aplicativos, que aprimoram a funcionalidade de servidores do World Wide Web (os computadores fornecem o conteúdo que é visualizado nos navegadores da Web), fornecem aplicativos para dispositivos destinados ao consumidor final (como telefones celulares, pagers e PDAs), entre outros. (DEITEL, 2010, p. 6).

2.3 JASPERREPORT

O JasperReport ⁵ é um framework para geração de relatórios. Entre as funcionalidades podemos destacar: É capaz de exportar relatórios para diversos formatos diferentes, tais como PDF, HTML, XML, XLS, etc; Aceita diversas formas de entrada de dados, tais como um arquivo XML ou CSV, conexão com o banco de dados, uma sessão do Hibernate, uma coleção de objetos em memória, etc; Permite o uso de diagramas, gráficos, e até códigos de barras. (<http://www.k19.com.br>)

2.4 IREPORT

O iReport⁶ é uma ferramenta desenvolvida pela mesma empresa do JasperReports, e por isso são utilizadas em conjuntos. Uma das dificuldades de trabalhar com relatórios, está na definição do layout. O iReport é um aplicativo gráfico, que permite que você “desenhe” um relatório, utilizando uma palheta, e arrastando e soltando componentes, simples e prática. (<http://www.k19.com.br/>).

⁵ www.jaspersoft.com

⁶ <http://www.k19.com.br/>

2.5 JAVA SERVE FACES (JSF)

Java Serve Faces (JSF)⁷ é uma tecnologia que nos permite criar aplicações Java Web utilizando componentes visuais, de forma que o desenvolvedor não se preocupe com Javascript e HTML. Basta adicionarmos os componentes (Tabelas, Calendários, Formulários, entre outros) e eles serão renderizados e exibidos em formato HTML. (<http://www.caelum.com.br/>)

2.6 HIBERNATE

O Hibernate é um framework de mapeamento objeto relacional para aplicações Java, ou seja, é uma ferramenta para mapear classes Java em tabelas do banco de dados e vice-versa. É bastante poderoso e dá suporte ao mapeamento de associações entre objetos.

2.7 ECLIPSE IDE

O Eclipse IDE⁸, chamado por muitos de Plataforma Eclipse, é considerada por uma grande maioria de desenvolvedores Java como a melhor ferramenta para desenvolvimento dessa linguagem. Desenvolvido inicialmente pela IBM e liberado o código fonte para a comunidade, criando a fundação Eclipse (*Eclipse Foundation*), sua grande vantagem, na atualidade, está no uso de seus assistentes de código e grande gama de *plug-ins*, podendo se adaptar a qualquer situação de desenvolvimento. (GONÇALVES, 2008, p. 3).

⁷ <http://www.caelum.com.br/>

⁸ <https://www.eclipse.org>

2.8 MYSQL

O MySQL⁹ é uma ferramenta gerenciadora de banco de dados, utilizada nas confecções de páginas web dinâmicas. A característica *open source* (código aberto) e o desempenho do conjunto popularizam o programa entre os desenvolvedores. As características do programa é que ele funciona em qualquer tipo de sistema operacional, outras características é a estabilidade e a baixa exigência de recursos de máquina. O MySQL é reconhecido por ser multitarefas e multiusuário e suporta vários tipos de tabelas, cada uma para uma finalidade específica. (<http://mysql.softonic.com.br/>).

⁹ <http://www.mysql.com/>

3 - ANÁLISE, ESPECIFICAÇÃO E PROJETO

Este capítulo está dividido em duas partes, na primeira parte do capítulo esta a análise e especificações do sistema, contemplando o levantamento de requisitos, lista de eventos, o caso de uso e suas especificações e os principais diagramas da UML. Na segunda parte do capítulo, está todo o planejamento do projeto.

3.1 ANÁLISE E ESPECIFICAÇÃO

3.1.1 Mapa mental

Os Mapas Mentais são métodos para organizar informações e tarefas, são diagramas que se constroem a partir de uma ideia central que se ramifica para outras ideias secundárias, usando imagens, cores e palavras-chave. Esse diagrama ajuda no raciocínio e compreensão das ideias sugeridas e nas tarefas que irão ser realizadas.

Para entendimento do sistema proposto, foi desenvolvido um mapa mental das funcionalidades do sistema, conforme a figura 1. Para a construção do Mapa Mental, foi utilizada a ferramenta FreeMind¹⁰, é uma ferramenta bastante útil para armazenar e organizar ideias.

¹⁰ www.freemind.sourceforge.net

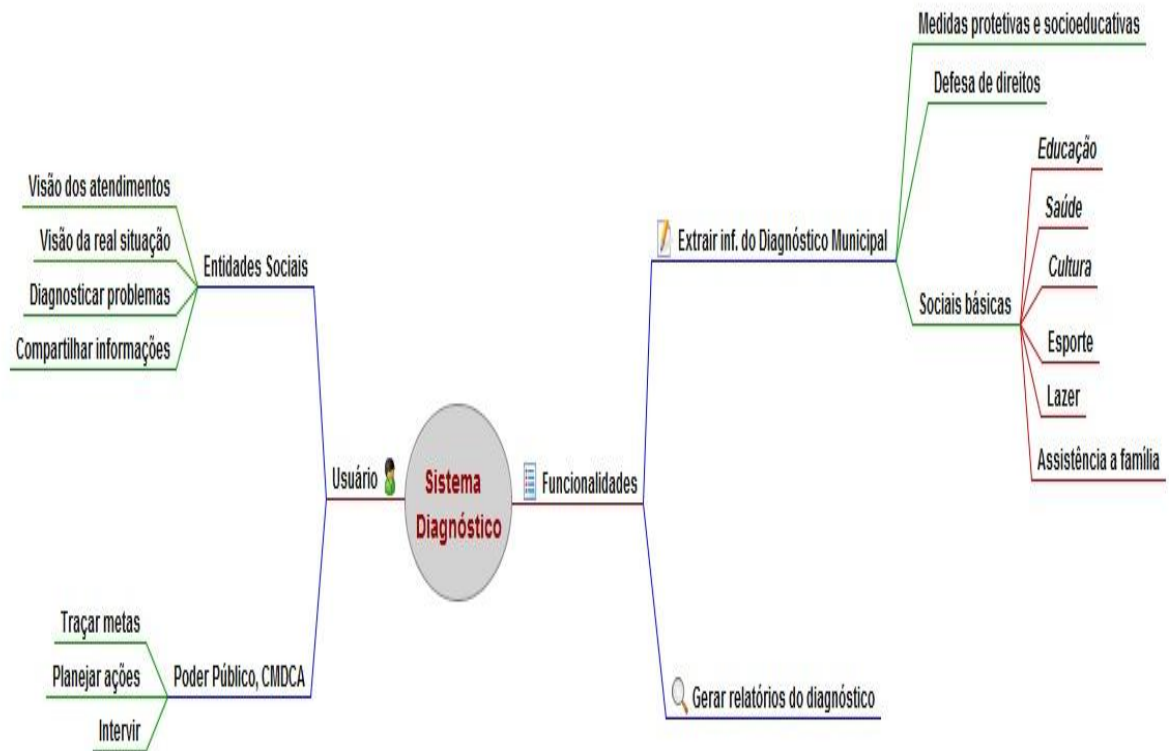


Figura 1- Mapa Mental das funcionalidades do sistema

O mapa mental abaixo apresenta a visão geral do sistema, com exemplos de relatórios.

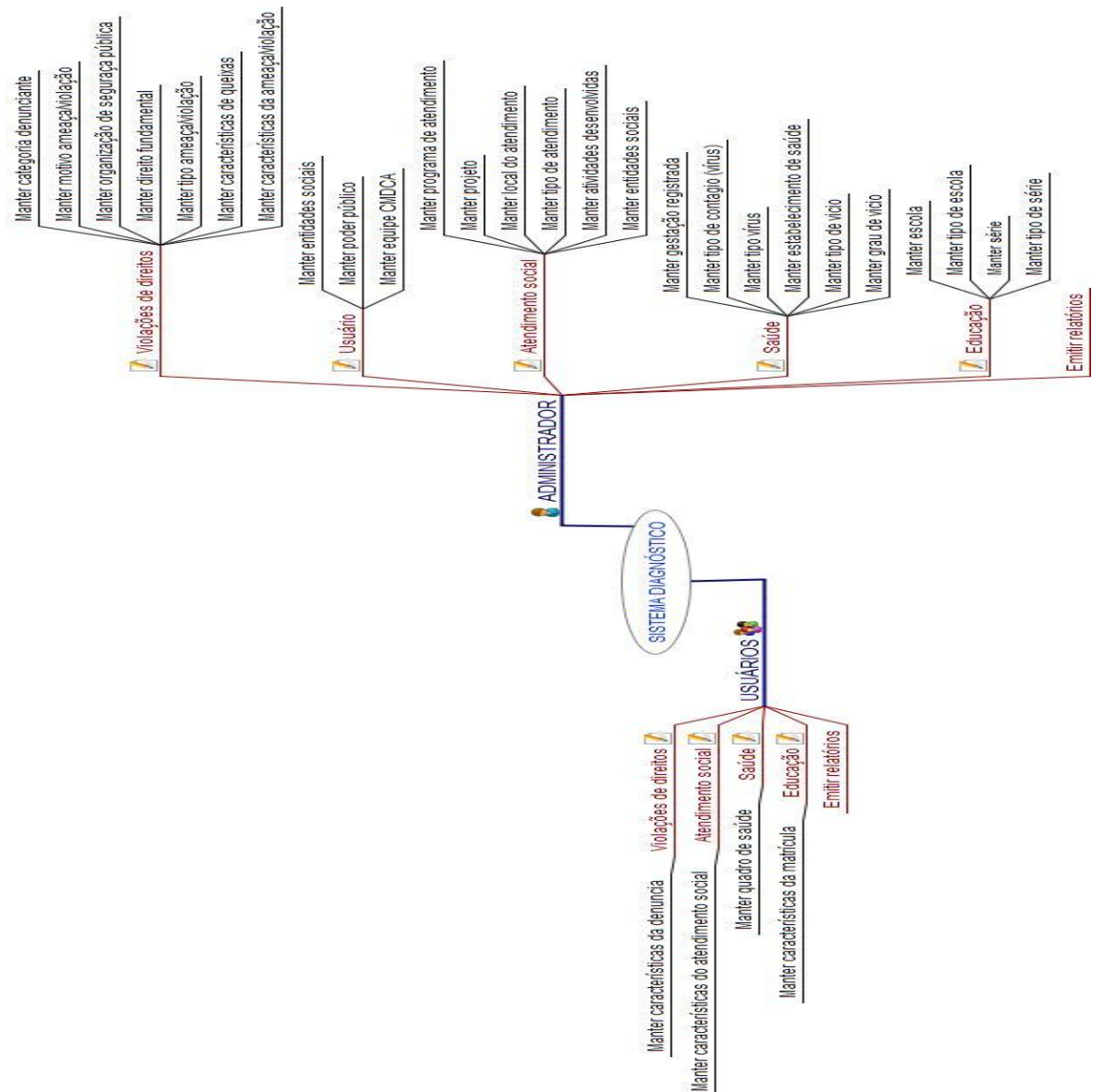


Figura 2- Mapa mental visão geral do sistema

3.1.2 Lista de eventos

Lista de Eventos			
Nº	Evento	Descrição	Ator
1	Efetuar Login	Efetuar login no Sistema.	Administrador/Usuário
2	Manter Usuários	Manter usuários no sistema.	Administrador
3	Manter categoria do denunciante	Manter tipo de categoria do denunciante.	Administrador
4	Manter motivo da ameaça ou violação	Manter motivo da ameaça ou violação ocorrida.	Administrador
5	Manter organização de segurança pública	Manter qual organização se refere os dados.	Administrador
6	Manter direito fundamental	Manter direito fundamental para cada tipo de violação.	Administrador
7	Manter tipo de ameaça/violação	Manter o tipo de ameaça/violação	Administrador
8	Manter características de queixas	Manter as características da queixa	Administrador
9	Manter características da ameaça/violação	Manter as características da ameaça/violação	Administrador
10	Manter características da denuncia	Manter as características da denuncia	Administrador/Usuário
11	Manter gestação	Manter gestações registradas	Administrador
12	Manter tipo de contágio	Manter qual foi o tipo de contágio adquirido pelo adolescente	Administrador
13	Manter tipo de vírus	Manter o tipo de vírus adquirido pelo adolescente	Administrador
14	Manter estabelecimento	Manter o estabelecimento assistencial de saúde	Administrador
15	Manter tipo de vício	Manter o tipo de vício do adolescente	Administrador
16	Manter grau de vício	Manter o nível do grau de vício do adolescente	Administrador

17	Manter quadro de saúde	Manter as características do quadro de saúde	Administrador/Usuário
18	Manter escola	Manter a escola que o adolescente frequenta	Administrador
19	Manter tipo de escola	Manter o tipo de escola	Administrador
20	Manter série	Manter a série do adolescente	Administrador
21	Manter tipo de série	Manter o tipo de série	Administrador
22	Manter características da matrícula	Manter as características das matrículas	Administrador/Usuário
23	Manter programa de atendimento	Manter os programas de atendimento aos adolescentes	Administrador
24	Manter projeto social	Manter o projeto pertencente ao adolescente	Administrador
25	Manter local do atendimento	Manter o local dos atendimentos sociais	Administrador
26	Manter tipo de atendimento	Manter o tipo de atendimento social	Administrador
27	Manter atividades	Manter as atividades desenvolvidas	Administrador
28	Manter entidade	Manter as entidades sociais	Administrador
29	Manter características do atendimento	Manter as características do atendimento social	Administrador/Usuário
30	Gerar relatórios	Gerar relatórios	Administrador/Usuário

Tabela 1- Lista de Eventos

3.1.3 Diagrama de casos de uso (UC)

A figura 3 apresenta o Diagrama de Caso de Uso, da visão geral do setor de segurança pública, com representação manter:

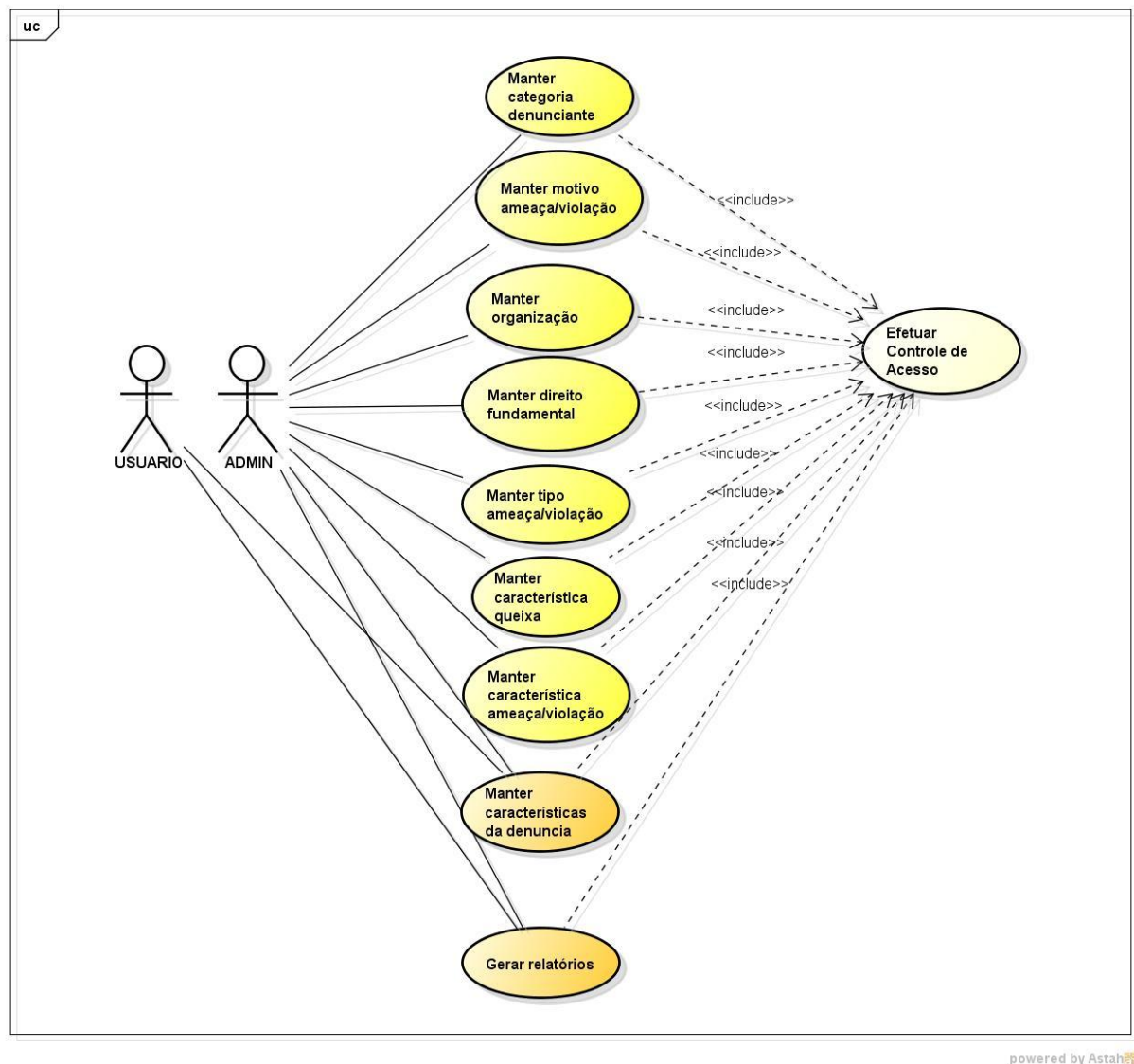


Figura 3-Segurança pública visão geral

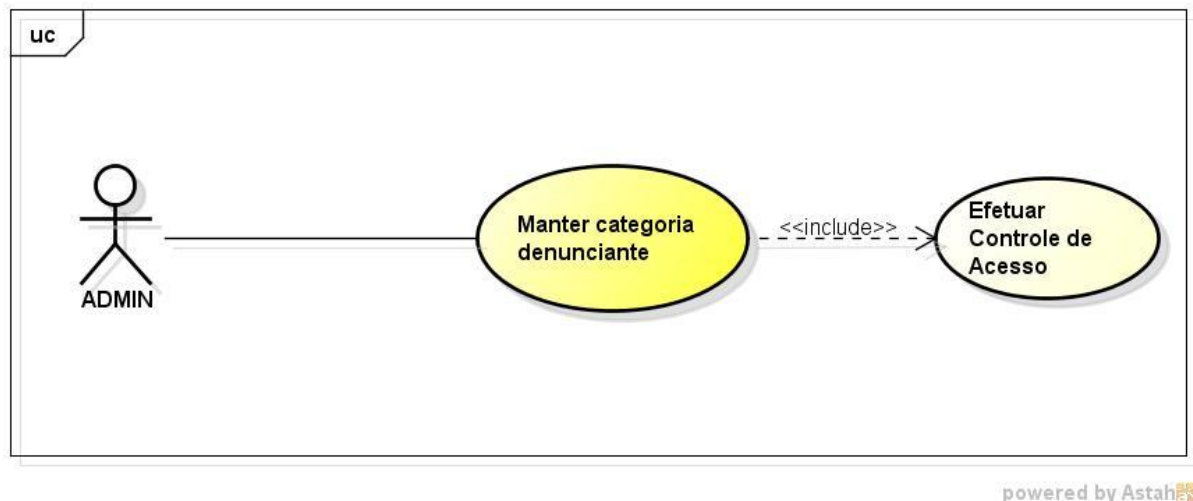


Figura 4- UC - Manter categoria denunciante

Nome do Caso de Uso	Manter categoria denunciante
Atores	Administrador
Pré-Condições	Efetuar Controle de Acesso
Cenário Principal	1- Administrador informa a categoria do denunciante e em seguida seleciona confirmar. 2- O sistema carrega os dados. 3- O Administrador finaliza a operação.
Cenário Alternativo	Administrador pode cancelar a operação durante o processo.
Casos de Teste	1- Os campos obrigatórios estando preenchidos o sistema executa a operação. 2- Caso contrário, o sistema retorna uma mensagem informando quais campos devem ser informados.

Tabela 2- Manter categoria denunciante

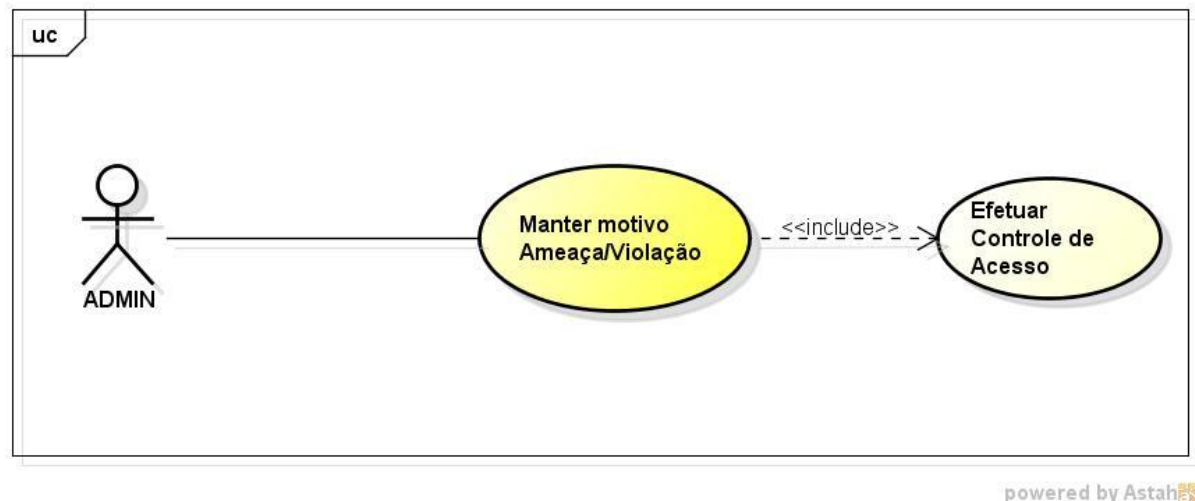


Figura 5 - UC - Manter motivo ameaça/violação

Nome do Caso de Uso	Manter motivo ameaça/violação
Atores	Administrador
Pré-Condições	Efetuar Controle de Acesso
Cenário Principal	1- Administrador informa o motivo da ameaça ou violação e em seguida seleciona confirmar. 2- O sistema carrega os dados. 3- O Administrador finaliza a operação.
Cenário Alternativo	Administrador pode cancelar a operação durante o processo.
Casos de Teste	1- Os campos obrigatórios estando preenchidos o sistema executa a operação. 2- Caso contrário, o sistema retorna uma mensagem informando quais campos devem ser informados.

Tabela 3- Manter motivo ameaça/violação

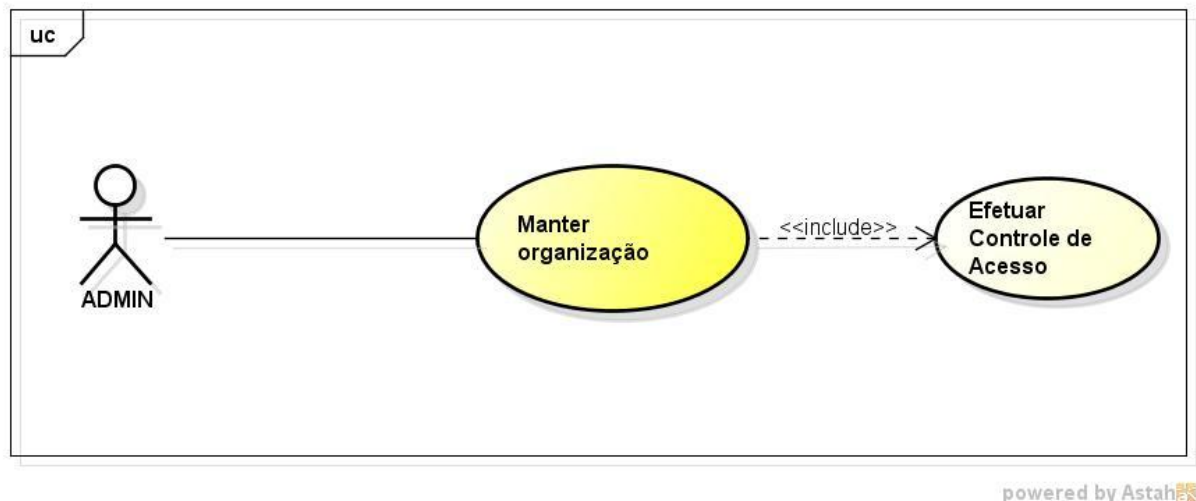


Figura 6- UC - Manter organização de segurança pública

Nome do Caso de Uso	Manter organização de segurança pública
Atores	Administrador
Pré-Condições	Efetuar Controle de Acesso
Cenário Principal	1- Administrador informa a organização e em seguida seleciona confirmar. 2- O sistema carrega os dados. 3- O Administrador finaliza a operação.
Cenário Alternativo	Administrador pode cancelar a operação durante o processo.
Casos de Teste	1- Os campos obrigatórios estando preenchidos o sistema executa a operação. 2- Caso contrário, o sistema retorna uma mensagem informando quais campos devem ser informados.

Tabela 4- Manter organização de segurança pública

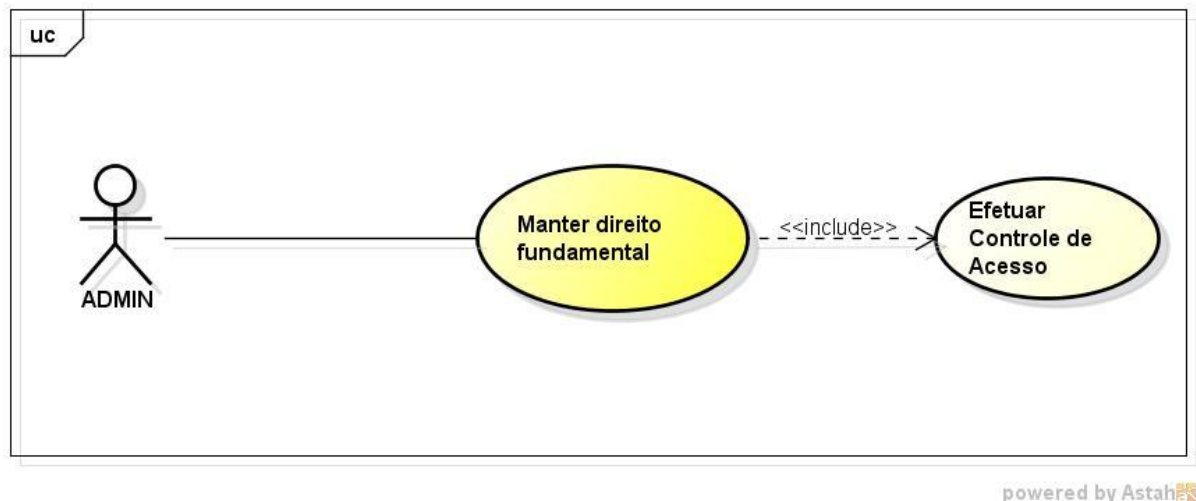


Figura 7- UC - Manter direito fundamental

Nome do Caso de Uso	Manter direito fundamental
Atores	Administrador
Pré-Condições	Efetuar Controle de Acesso
Cenário Principal	1- Administrador informa o direito fundamental e em seguida seleciona confirmar. 2- O sistema carrega os dados. 3- O Administrador finaliza a operação.
Cenário Alternativo	Administrador pode cancelar a operação durante o processo.
Casos de Teste	1- Os campos obrigatórios estando preenchidos o sistema executa a operação. 2- Caso contrário, o sistema retorna uma mensagem informando quais campos devem ser informados.

Tabela 5- Manter direito fundamental

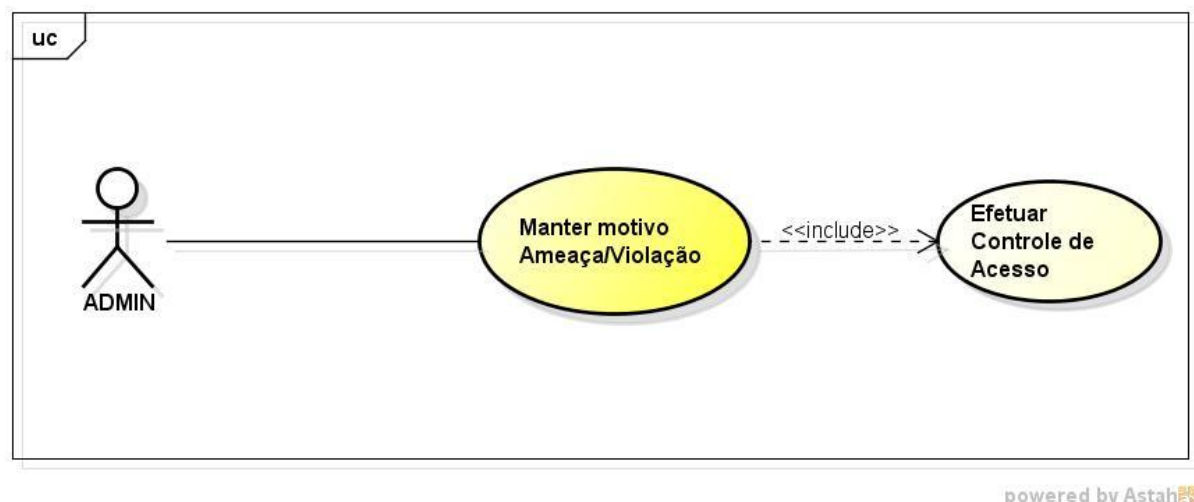


Figura 8- UC - Manter tipo ameaça/violação

Nome do Caso de Uso	Manter tipo ameaça/violação
Atores	Administrador
Pré-Condições	Efetuar Controle de Acesso
Cenário Principal	1- Administrador informa o tipo da ameaça/violação e em seguida seleciona confirmar. 2- O sistema carrega os dados. 3- O Administrador finaliza a operação.
Cenário Alternativo	Administrador pode cancelar a operação durante o processo.
Casos de Teste	1- Os campos obrigatórios estando preenchidos o sistema executa a operação. 2- Caso contrário, o sistema retorna uma mensagem informando quais campos devem ser informados.

Tabela 6- Manter tipo ameaça/violação

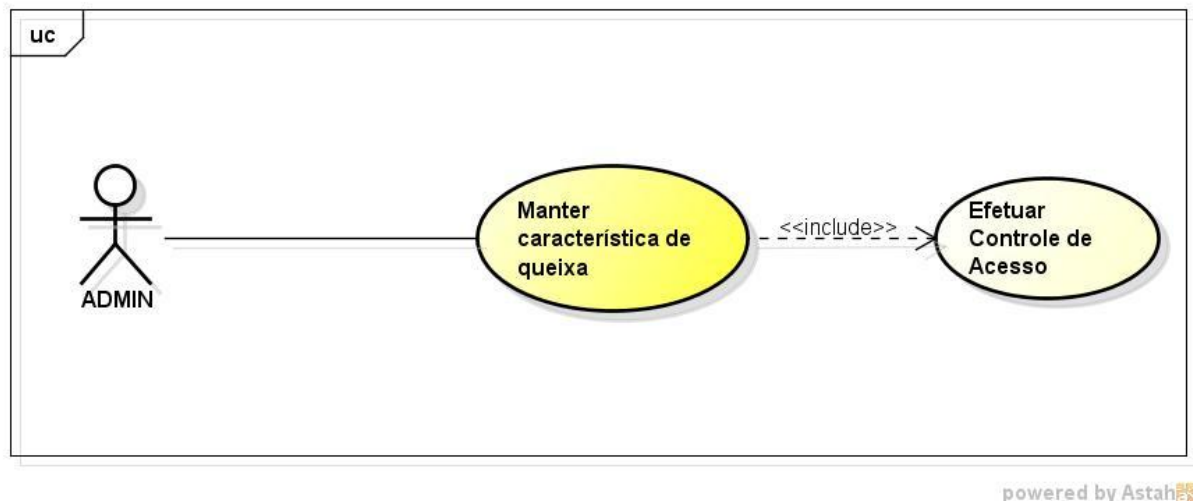


Figura 9- UC - Manter característica da queixa

Nome do Caso de Uso	Manter característica da queixa
Atores	Administrador
Pré-Condições	Efetuar Controle de Acesso
Cenário Principal	1- Administrador informa a característica da queixa e em seguida seleciona confirmar. 2- O sistema carrega os dados. 3- O Administrador finaliza a operação.
Cenário Alternativo	Administrador pode cancelar a operação durante o processo.
Casos de Teste	1- Os campos obrigatórios estando preenchidos o sistema executa a operação. 2- Caso contrário, o sistema retorna uma mensagem informando quais campos devem ser informados.

Tabela 7- Manter característica da queixa

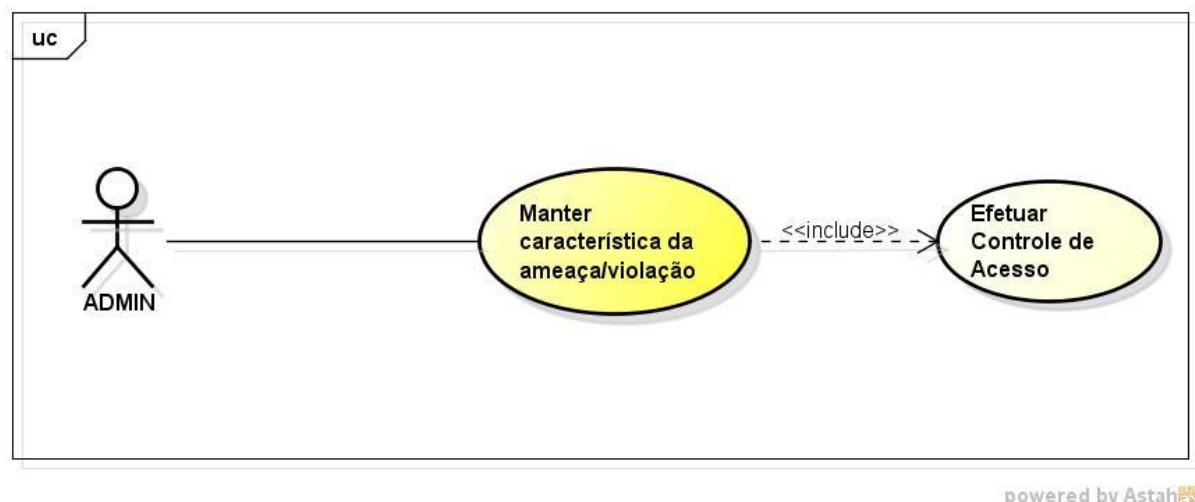


Figura 10-UC - Manter característica da ameaça/violação

Nome do Caso de Uso	Manter característica da ameaça/violação
Atores	Administrador
Pré-Condições	Efetuar Controle de Acesso
Cenário Principal	1- Administrador informa a característica da ameaça/violação e em seguida seleciona confirmar. 2- O sistema carrega os dados. 3- O Administrador finaliza a operação.
Cenário Alternativo	Administrador pode cancelar a operação durante o processo.
Casos de Teste	1- Os campos obrigatórios estando preenchidos o sistema executa a operação. 2- Caso contrário, o sistema retorna uma mensagem informando quais campos devem ser informados.

Tabela 8-Manter característica da ameaça/violação

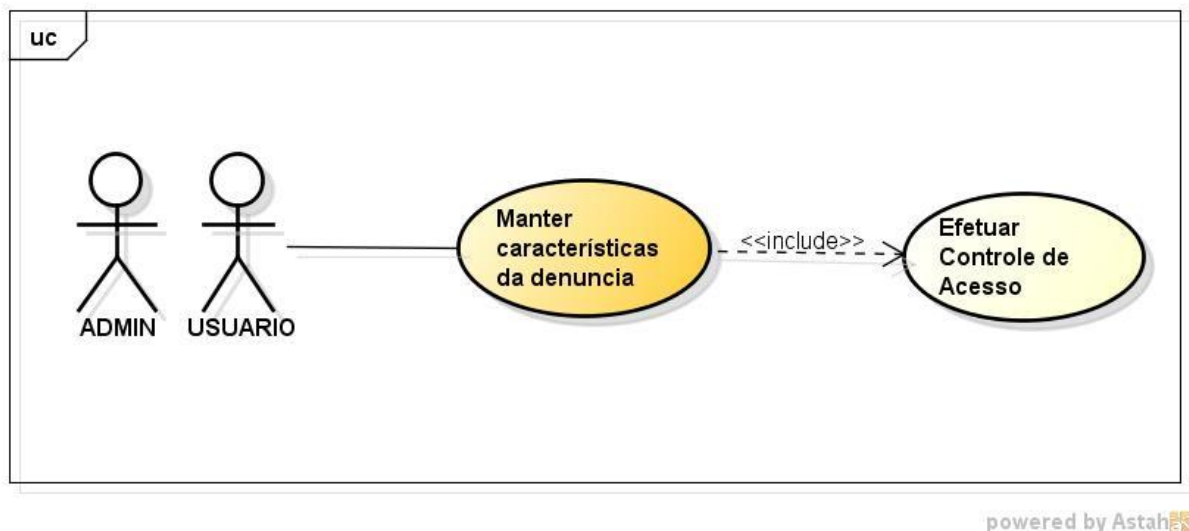
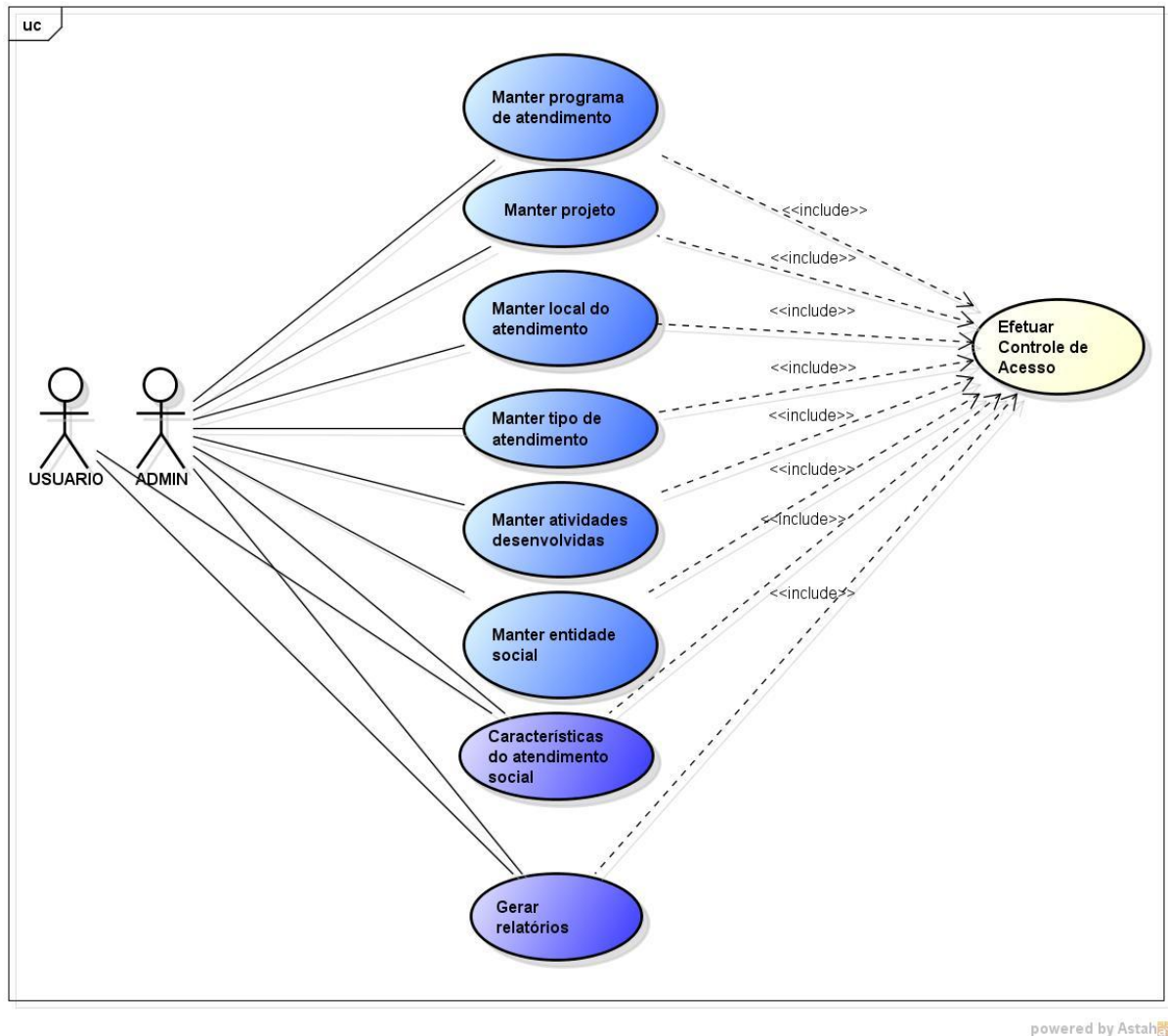


Figura 11- UC - Manter características da denuncia

Nome do Caso de Uso	Manter características da denuncia
Atores	Administrador/Usuário
Pré-Condições	Efetuar Controle de Acesso
Cenário Principal	4- Administrador/Usuário informa as características da denuncia e em seguida seleciona confirmar. 5- O sistema carrega os dados. 6- O Administrador/Usuário finaliza a operação.
Cenário Alternativo	Administrador/Usuário pode cancelar a operação durante o processo.
Casos de Teste	3- Os campos obrigatórios estando preenchidos o sistema executa a operação. 4- Caso contrário, o sistema retorna uma mensagem informando quais campos devem ser informados.

Tabela 9- Manter características da denuncia

A figura 12 apresenta o Diagrama de Caso de Uso, da visão geral dos atendimentos sociais, com representação manter:



powered by Astah

Figura 12- UC - Atendimento social visão geral

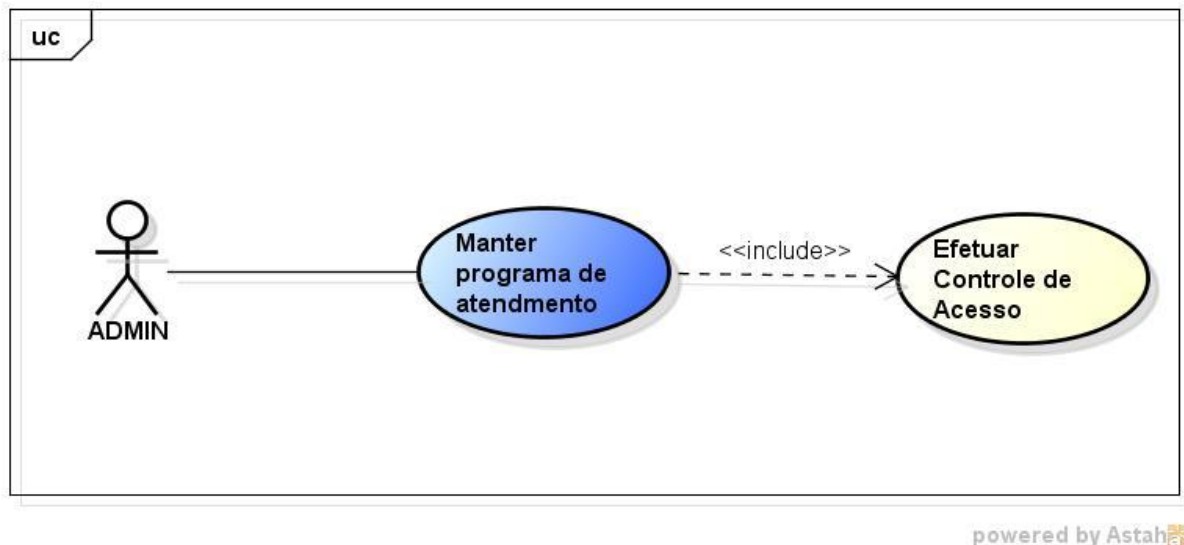


Figura 13- UC - Manter programa de atendimento

Nome do Caso de Uso	Manter programa de atendimento
Atores	Administrador
Pré-Condições	Efetuar Controle de Acesso
Cenário Principal	1- Administrador informa o programa de atendimento e em seguida seleciona confirmar 2- O sistema carrega os dados. 3- O Administrador finaliza a operação.
Cenário Alternativo	Administrador pode cancelar a operação durante o processo.
Casos de Teste	1- Os campos obrigatórios estando preenchidos o sistema executa a operação. 2- Caso contrário, o sistema retorna uma mensagem informando quais campos devem ser informados.

Tabela 10- Manter programa de atendimento

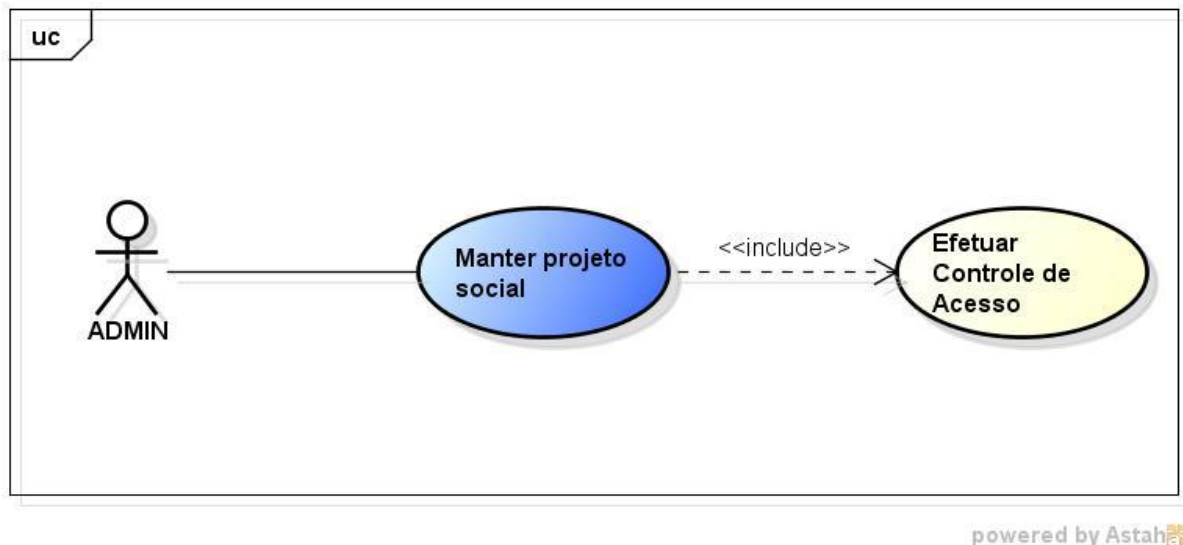


Figura 14- UC - Manter projeto social

Nome do Caso de Uso	Manter projeto social
Atores	Administrador
Pré-Condições	Efetuar Controle de Acesso
Cenário Principal	1- Administrador informa o projeto social e em seguida seleciona confirmar. 2- O sistema carrega os dados. 3- O Administrador finaliza a operação.
Cenário Alternativo	Administrador pode cancelar a operação durante o processo.
Casos de Teste	1- Os campos obrigatórios estando preenchidos o sistema executa a operação. 2- Caso contrário, o sistema retorna uma mensagem informando quais campos devem ser informados.

Tabela 11- Manter projeto social

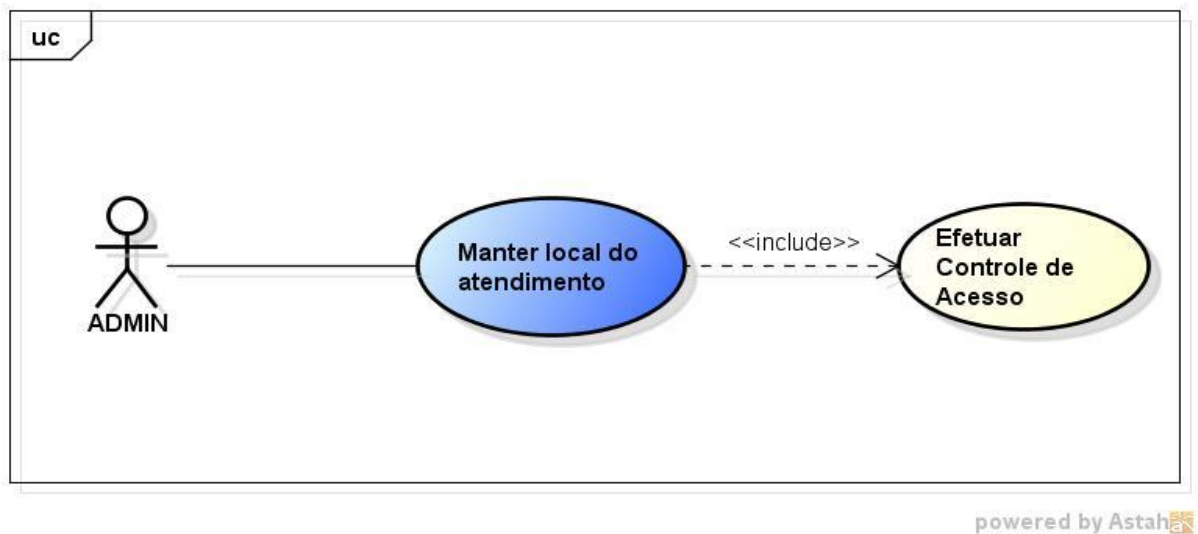


Figura 15- UC - Manter local do atendimento

Nome do Caso de Uso	Manter local do atendimento
Atores	Administrador
Pré-Condições	Efetuar Controle de Acesso
Cenário Principal	1- Administrador informa o local do atendimento e em seguida seleciona confirmar. 2- O sistema carrega os dados. 3- O Administrador finaliza a operação.
Cenário Alternativo	Administrador pode cancelar a operação durante o processo.
Casos de Teste	1- Os campos obrigatórios estando preenchidos o sistema executa a operação. 2- Caso contrário, o sistema retorna uma mensagem informando quais campos devem ser informados.

Tabela 12- Manter local do atendimento

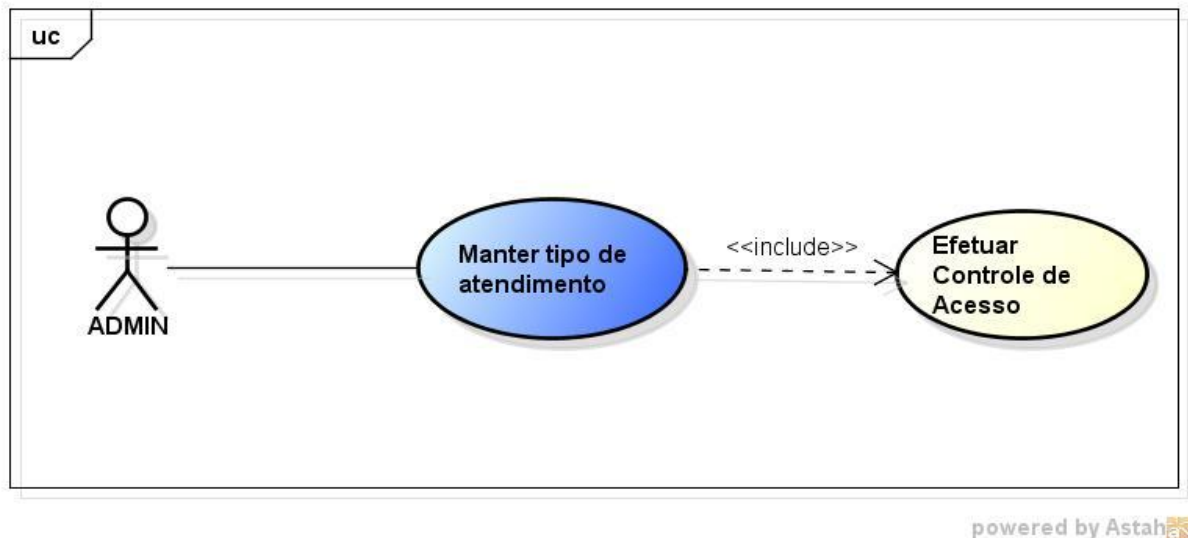


Figura 16- UC - Manter tipo de atendimento

Nome do Caso de Uso	Manter tipo de atendimento
Atores	Administrador
Pré-Condições	Efetuar Controle de Acesso
Cenário Principal	1- Administrador informa o tipo de atendimento e em seguida seleciona confirmar. 2- O sistema carrega os dados. 3- O Administrador finaliza a operação.
Cenário Alternativo	Administrador pode cancelar a operação durante o processo.
Casos de Teste	1- Os campos obrigatórios estando preenchidos o sistema executa a operação. 2- Caso contrário, o sistema retorna uma mensagem informando quais campos devem ser informados.

Tabela 13- Manter tipo de atendimento

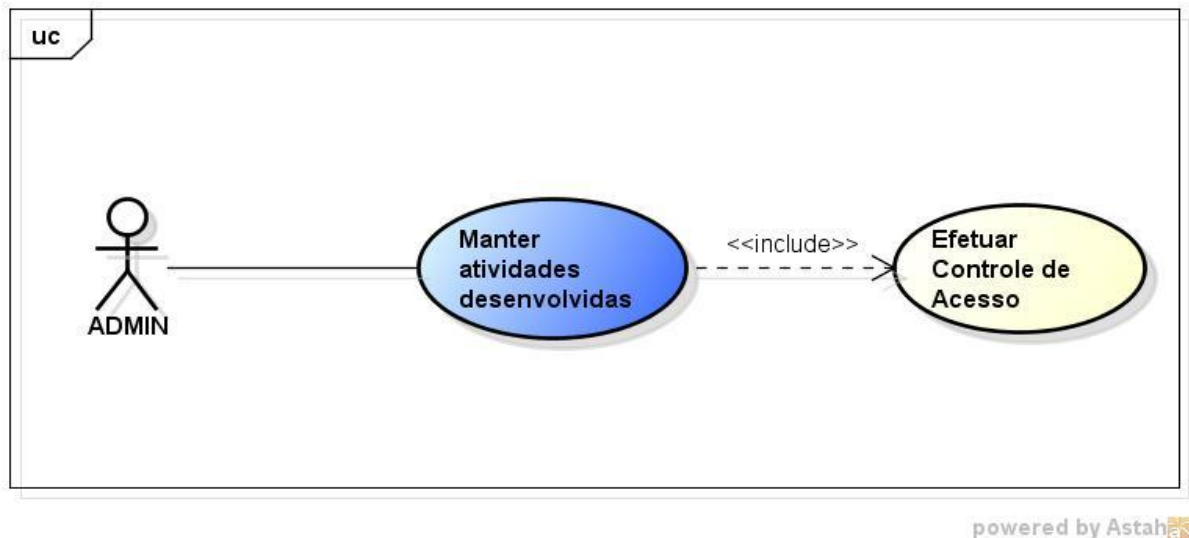


Figura 17- UC - Manter atividades desenvolvidas

Nome do Caso de Uso	Manter atividades desenvolvidas
Atores	Administrador
Pré-Condições	Efetuar Controle de Acesso
Cenário Principal	1- Administrador informa a atividade desenvolvida e em seguida seleciona confirmar. 2- O sistema carrega os dados. 3- O Administrador finaliza a operação.
Cenário Alternativo	Administrador pode cancelar a operação durante o processo.
Casos de Teste	1- Os campos obrigatórios estando preenchidos o sistema executa a operação. 2- Caso contrário, o sistema retorna uma mensagem informando quais campos devem ser informados.

Tabela 14- Manter atividades desenvolvidas

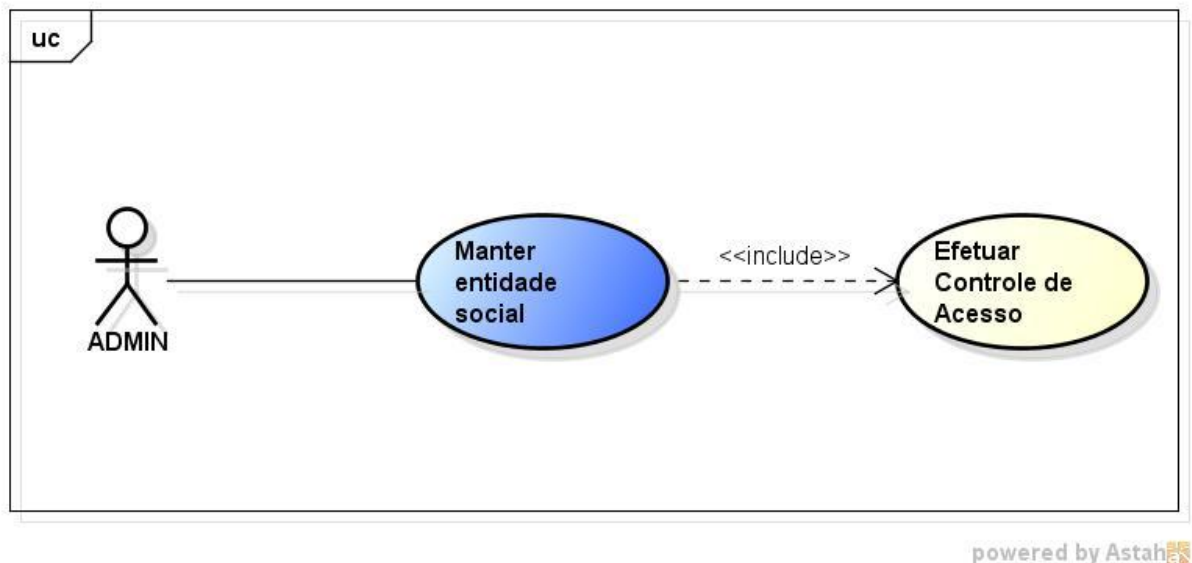


Figura 18- UC - Manter entidade social

Nome do Caso de Uso	Manter entidade social
Atores	Administrador
Pré-Condições	Efetuar Controle de Acesso
Cenário Principal	1- Administrador informa a entidade social e em seguida seleciona confirmar. 2- O sistema carrega os dados. 3- O Administrador finaliza a operação.
Cenário Alternativo	Administrador pode cancelar a operação durante o processo.
Casos de Teste	1- Os campos obrigatórios estando preenchidos o sistema executa a operação. 2- Caso contrário, o sistema retorna uma mensagem informando quais campos devem ser informados.

Tabela 15- Manter entidade social

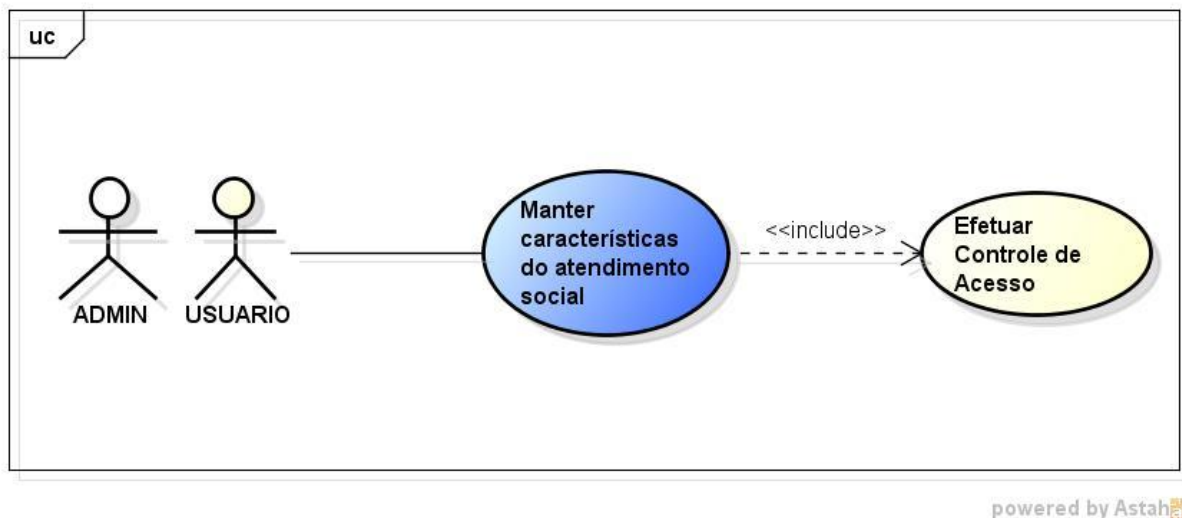


Figura 19- UC - Manter características do atendimento social

Nome do Caso de Uso	Manter características do atendimento social
Atores	Administrador/Usuário
Pré-Condições	Efetuar Controle de Acesso
Cenário Principal	1- Administrador/Usuário informa as características do atendimento social e em seguida seleciona confirmar. 2- O sistema carrega os dados. 3- O Administrador/Usuário finaliza a operação.
Cenário Alternativo	Administrador/Usuário pode cancelar a operação durante o processo.
Casos de Teste	1- Os campos obrigatórios estando preenchidos o sistema executa a operação. 2- Caso contrário, o sistema retorna uma mensagem informando quais campos devem ser informados.

Tabela 16- Manter características do atendimento social

A figura 20 apresenta o Diagrama de Caso de Uso, da visão geral do setor da saúde, com representação manter:

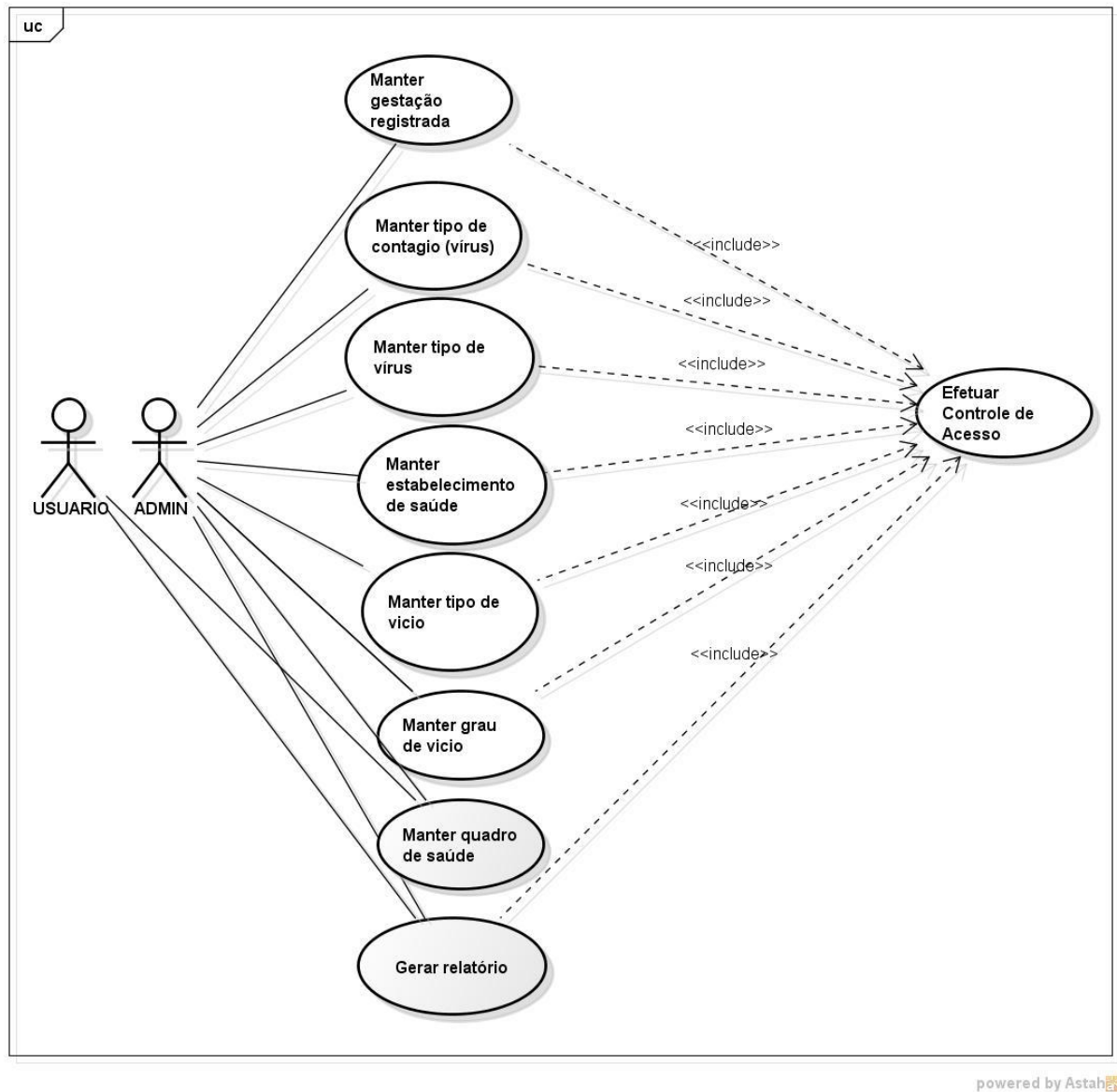


Figura 20- UC - Setor da saúde visão geral

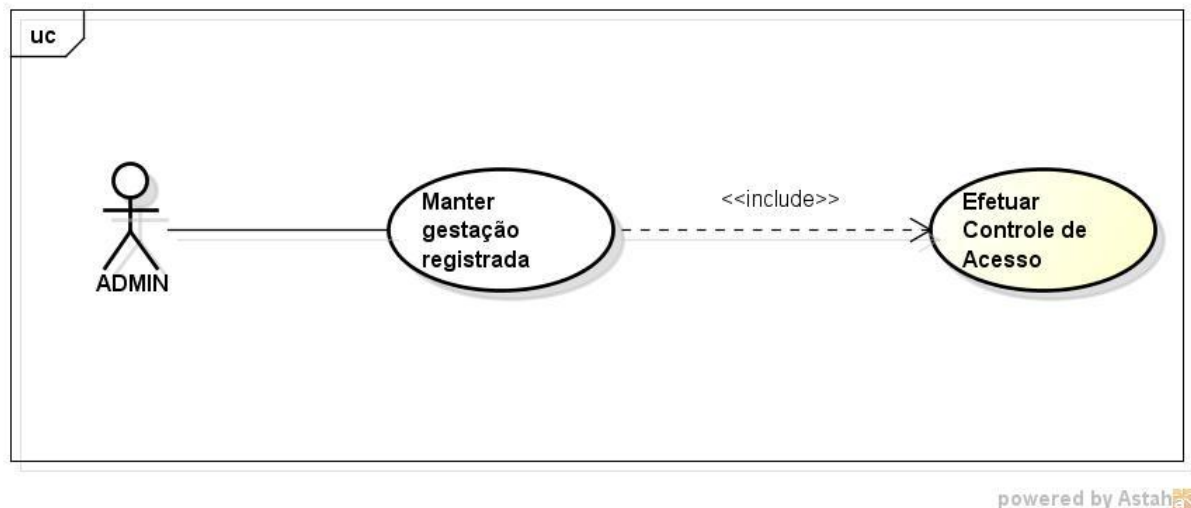
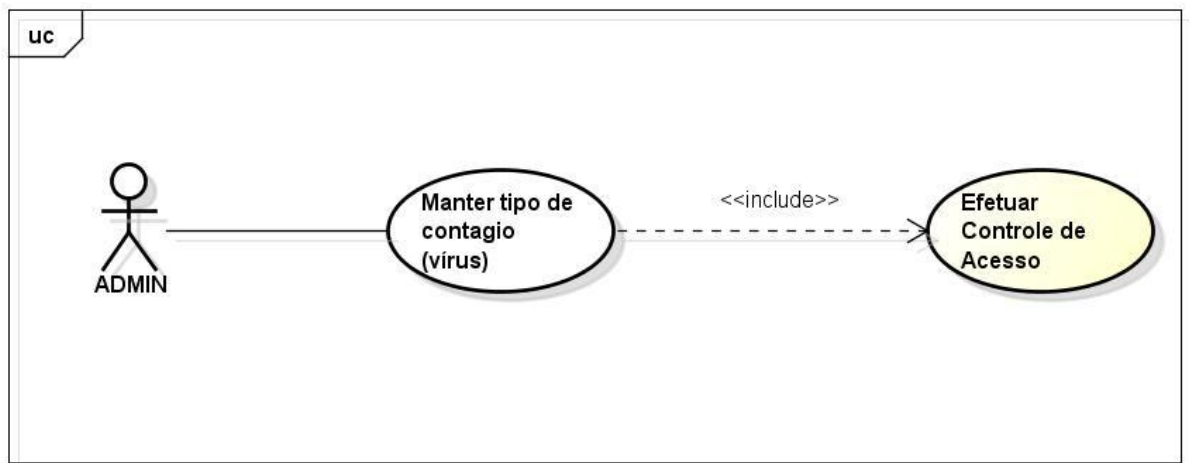


Figura 21- UC - Manter gestação registrada

Nome do Caso de Uso	Manter gestação registrada
Atores	Administrador
Pré-Condições	Efetuar Controle de Acesso
Cenário Principal	1- Administrador informa a gestação registrada e em seguida seleciona confirmar. 2- O sistema carrega os dados. 3- O Administrador finaliza a operação.
Cenário Alternativo	Administrador pode cancelar a operação durante o processo.
Casos de Teste	1- Os campos obrigatórios estando preenchidos o sistema executa a operação. 2- Caso contrário, o sistema retorna uma mensagem informando quais campos devem ser informados.

Tabela 17- Manter gestação registrada



powered by Astah

Figura 22- UC - Manter tipo de contágio (vírus)

Nome do Caso de Uso	Manter tipo de contágio (vírus)
Atores	Administrador
Pré-Condições	Efetuar Controle de Acesso
Cenário Principal	1- Administrador informa o tipo de contágio (vírus) e em seguida seleciona confirmar. 2- O sistema carrega os dados. 3- O Administrador finaliza a operação.
Cenário Alternativo	Administrador pode cancelar a operação durante o processo.
Casos de Teste	1- Os campos obrigatórios estando preenchidos o sistema executa a operação. 2- Caso contrário, o sistema retorna uma mensagem informando quais campos devem ser informados.

Tabela 18- Manter tipo de contágio (vírus)

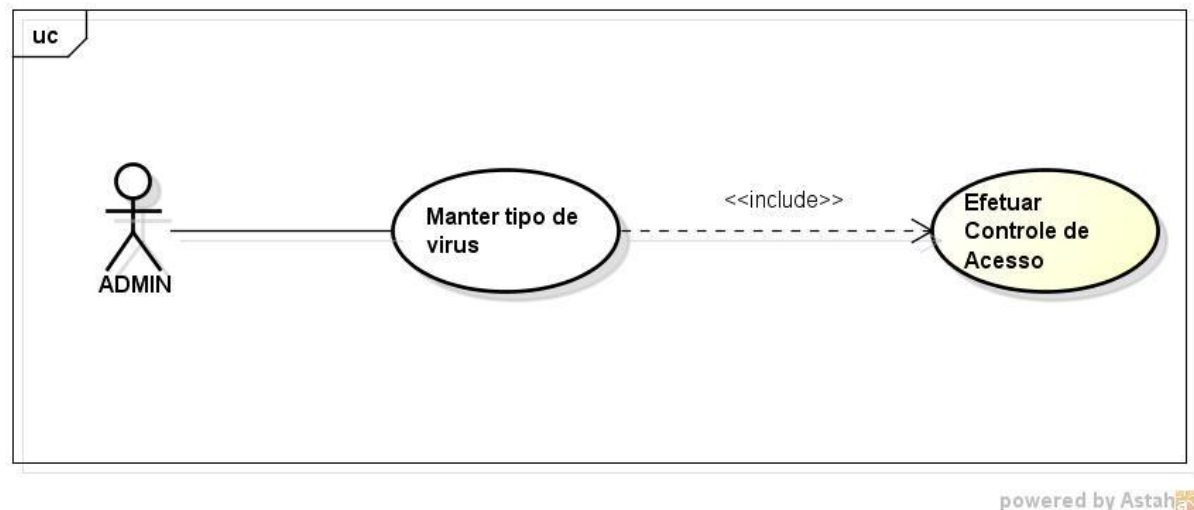


Figura 23- UC - Manter tipo de vírus

Nome do Caso de Uso	Manter tipo de vírus
Atores	Administrador
Pré-Condições	Efetuar Controle de Acesso
Cenário Principal	1- Administrador informa o tipo de vírus e em seguida seleciona confirmar. 2- O sistema carrega os dados. 3- O Administrador finaliza a operação.
Cenário Alternativo	Administrador pode cancelar a operação durante o processo.
Casos de Teste	1- Os campos obrigatórios estando preenchidos o sistema executa a operação. 2- Caso contrário, o sistema retorna uma mensagem informando quais campos devem ser informados.

Tabela 19- Manter tipo de vírus

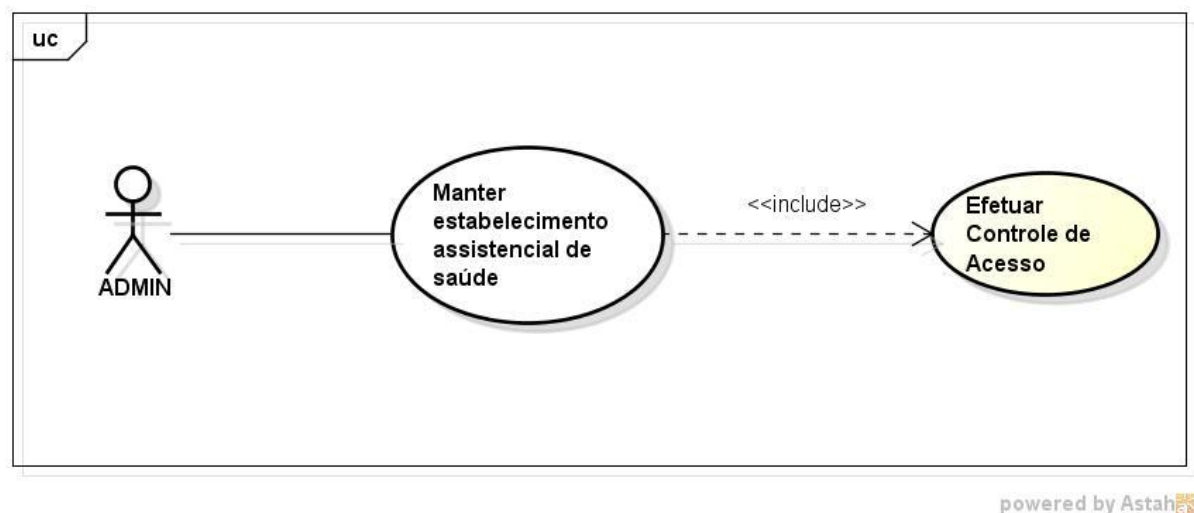
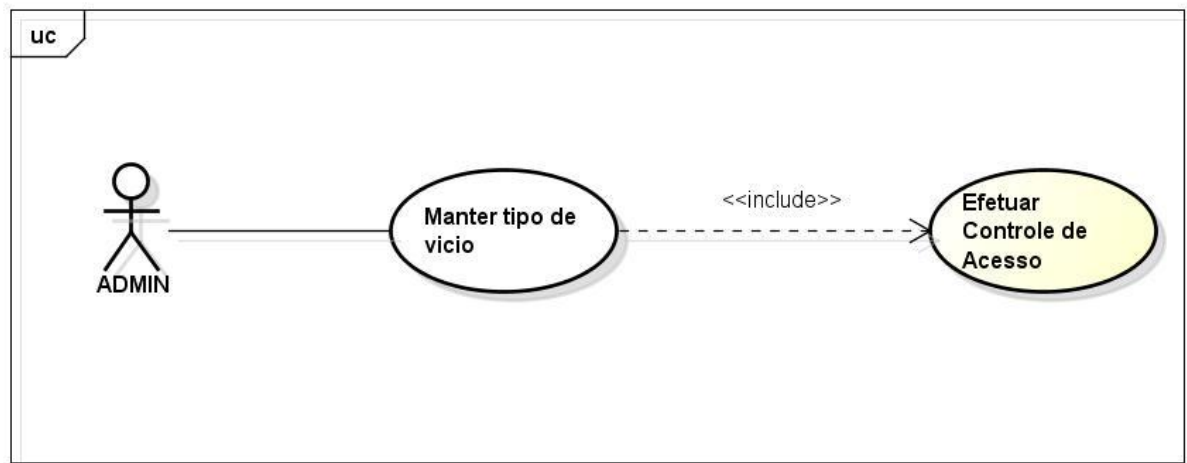


Figura 24- UC - Manter estabelecimento assistencial de saúde

Nome do Caso de Uso	Manter estabelecimento assistencial de saúde
Atores	Administrador
Pré-Condições	Efetuar Controle de Acesso
Cenário Principal	1- Administrador informa o estabelecimento de saúde e em seguida seleciona confirmar. 2- O sistema carrega os dados. 3- O Administrador finaliza a operação.
Cenário Alternativo	Administrador pode cancelar a operação durante o processo.
Casos de Teste	1- Os campos obrigatórios estando preenchidos o sistema executa a operação. 2- Caso contrário, o sistema retorna uma mensagem informando quais campos devem ser informados.

Tabela 20- Manter estabelecimento assistencial de saúde



powered by Astah

Figura 25- UC - Manter tipo de vício

Nome do Caso de Uso	Manter tipo de vício
Atores	Administrador
Pré-Condições	Efetuar Controle de Acesso
Cenário Principal	1- Administrador informa o tipo de vício e em seguida seleciona confirmar. 2- O sistema carrega os dados. 3- O Administrador finaliza a operação.
Cenário Alternativo	Administrador pode cancelar a operação durante o processo.
Casos de Teste	1- Os campos obrigatórios estando preenchidos o sistema executa a operação. 2- Caso contrário, o sistema retorna uma mensagem informando quais campos devem ser informados.

Tabela 21- Manter tipo de vício

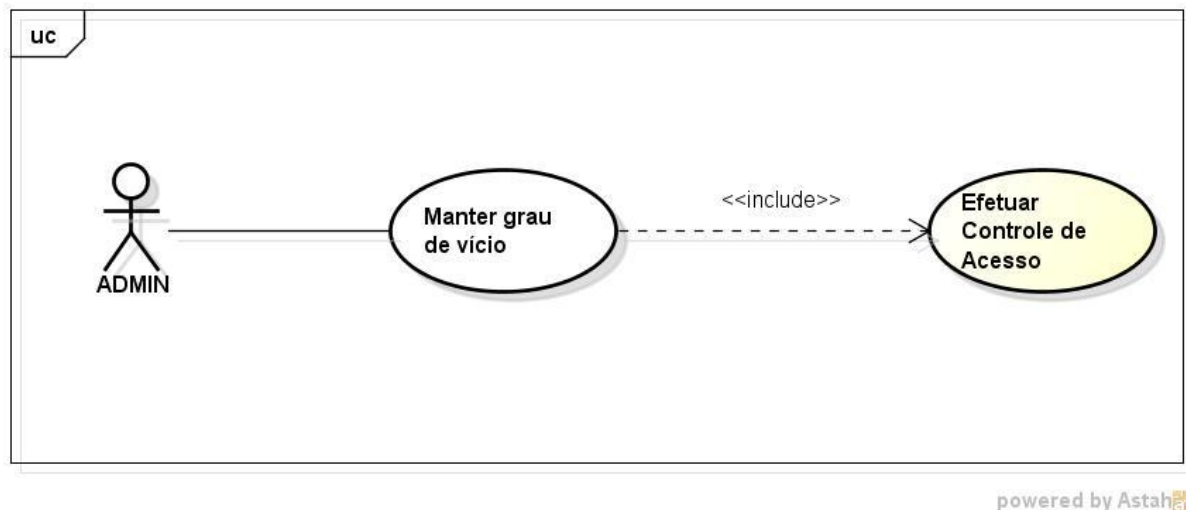


Figura 26- UC - Manter grau de vício

Nome do Caso de Uso	Manter grau de vício
Atores	Administrador
Pré-Condições	Efetuar Controle de Acesso
Cenário Principal	1- Administrador informa o grau de vício e em seguida seleciona confirmar. 2- O sistema carrega os dados. 3- O Administrador finaliza a operação.
Cenário Alternativo	Administrador pode cancelar a operação durante o processo.
Casos de Teste	1- Os campos obrigatórios estando preenchidos o sistema executa a operação. 2- Caso contrário, o sistema retorna uma mensagem informando quais campos devem ser informados.

Tabela 22- Manter grau de vício

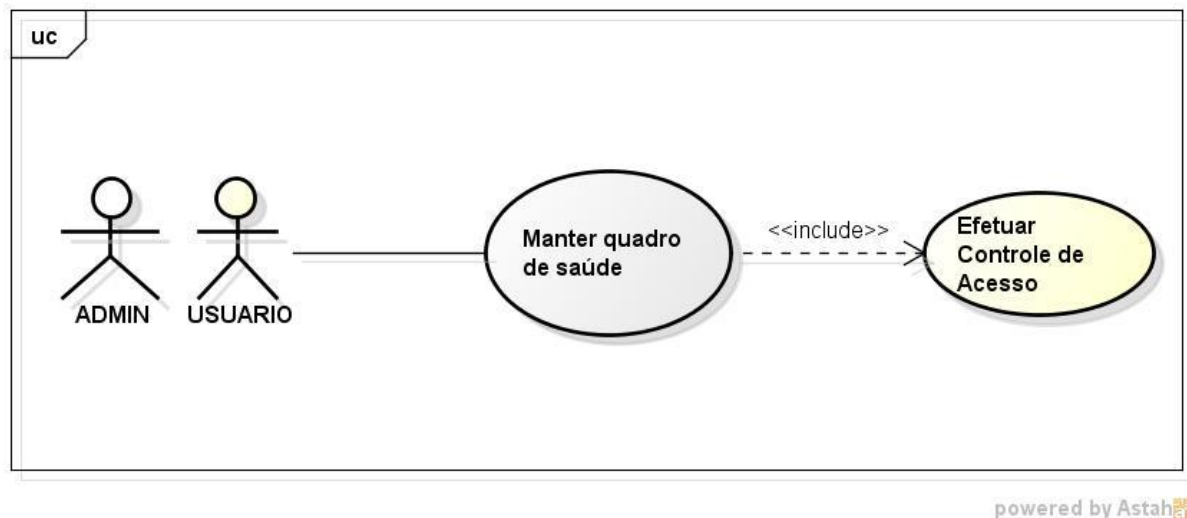


Figura 27- UC - Manter quadro de saúde

Nome do Caso de Uso	Manter quadro de saúde
Atores	Administrador/Usuário
Pré-Condições	Efetuar Controle de Acesso
Cenário Principal	1- Administrador/Usuário informa o quadro de saúde e em seguida seleciona confirmar. 2- O sistema carrega os dados. 3- O Administrador/Usuário finaliza a operação.
Cenário Alternativo	Administrador/Usuário pode cancelar a operação durante o processo.
Casos de Teste	1- Os campos obrigatórios estando preenchidos o sistema executa a operação. 2- Caso contrário, o sistema retorna uma mensagem informando quais campos devem ser informados.

Tabela 23- Manter quadro de saúde

A figura 28 apresenta o Diagrama de Caso de Uso, da visão geral do setor da educação, com representação manter:

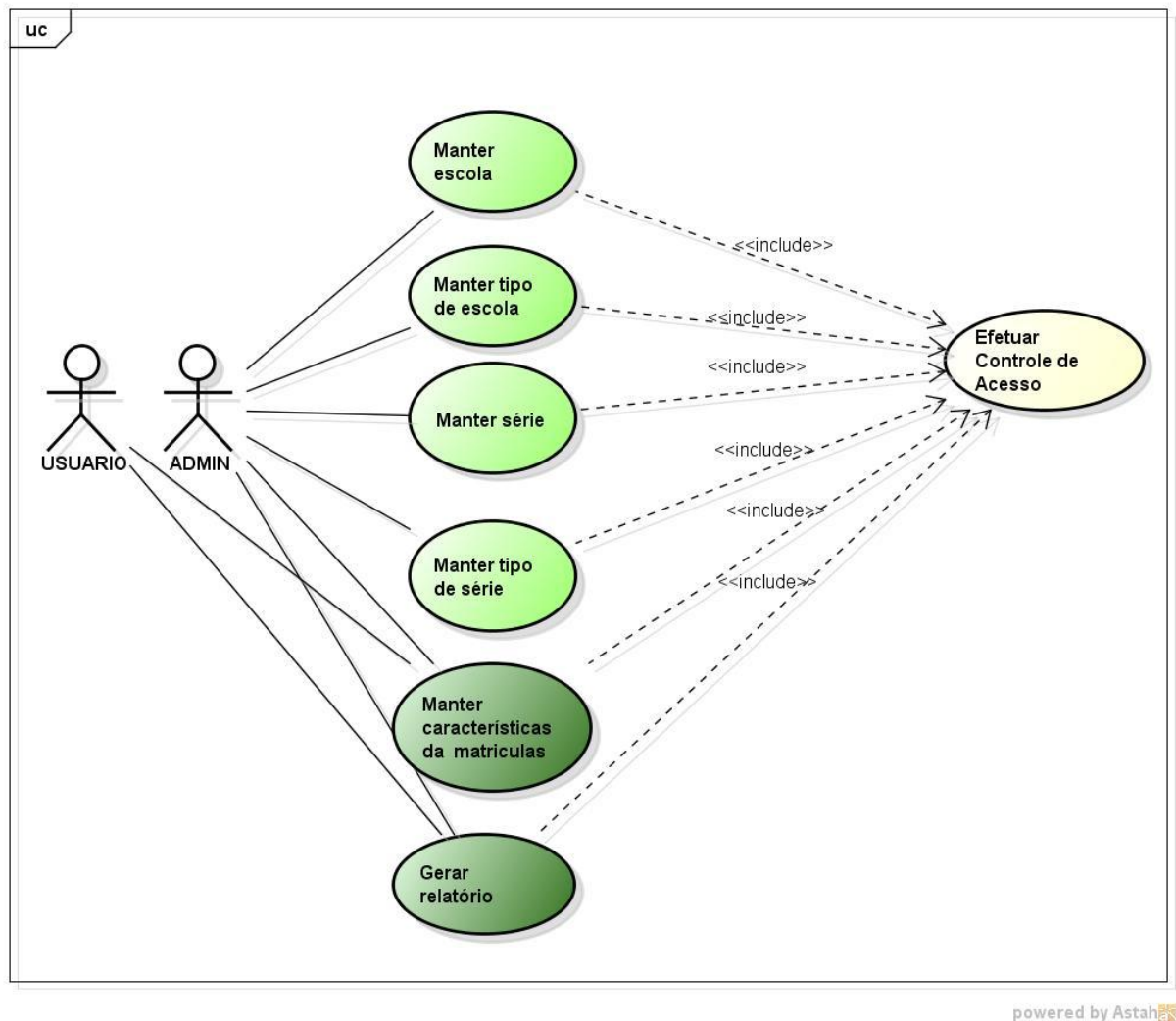


Figura 28- UC - Setor da educação visão geral

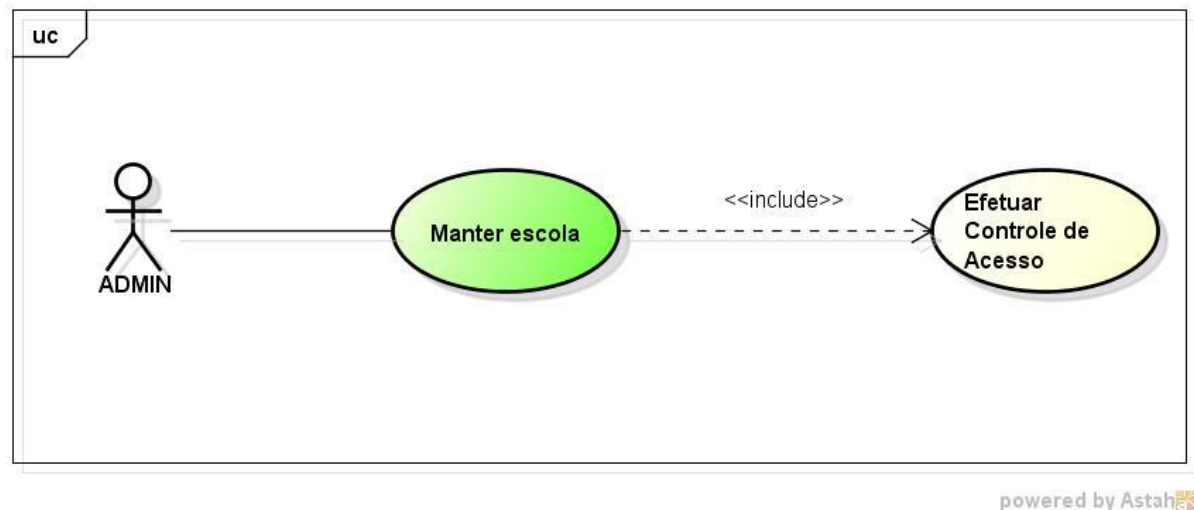


Figura 29- UC - Manter escola

Nome do Caso de Uso	Manter escola
Atores	Administrador
Pré-Condições	Efetuar Controle de Acesso
Cenário Principal	1- Administrador informa a escola e em seguida seleciona confirmar. 2- O sistema carrega os dados. 3- O Administrador finaliza a operação.
Cenário Alternativo	Administrador pode cancelar a operação durante o processo.
Casos de Teste	1- Os campos obrigatórios estando preenchidos o sistema executa a operação. 2- Caso contrário, o sistema retorna uma mensagem informando quais campos devem ser informados.

Tabela 24- Manter escola

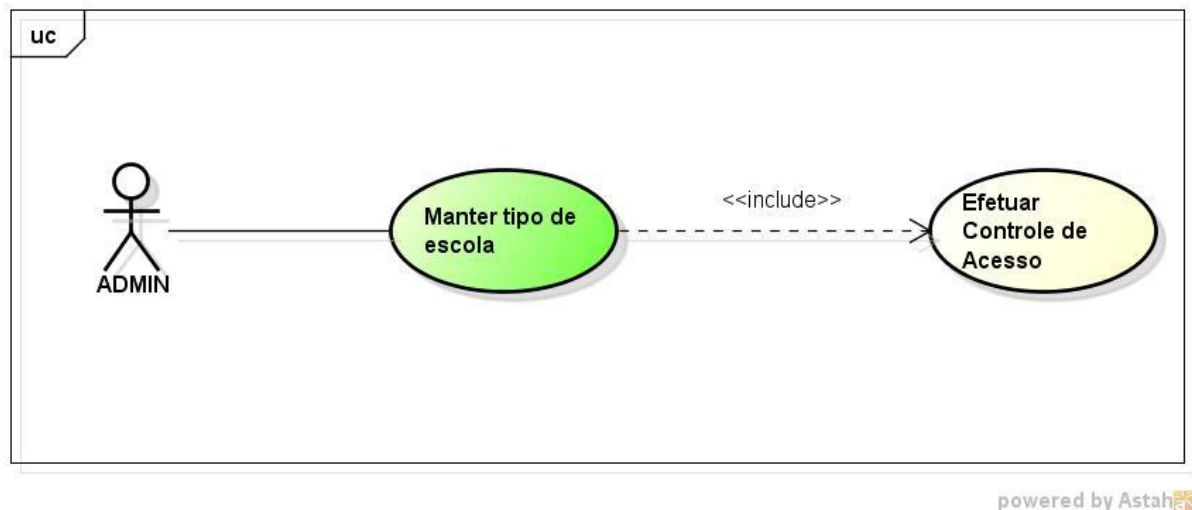


Figura 30- UC - Manter tipo de escola

Nome do Caso de Uso	Manter tipo de escola
Atores	Administrador
Pré-Condições	Efetuar Controle de Acesso
Cenário Principal	1- Administrador informa o tipo de escola e em seguida seleciona confirmar. 2- O sistema carrega os dados. 3- O Administrador finaliza a operação.
Cenário Alternativo	Administrador pode cancelar a operação durante o processo.
Casos de Teste	1- Os campos obrigatórios estando preenchidos o sistema executa a operação. 2- Caso contrário, o sistema retorna uma mensagem informando quais campos devem ser informados.

Tabela 25- Manter tipo de escola

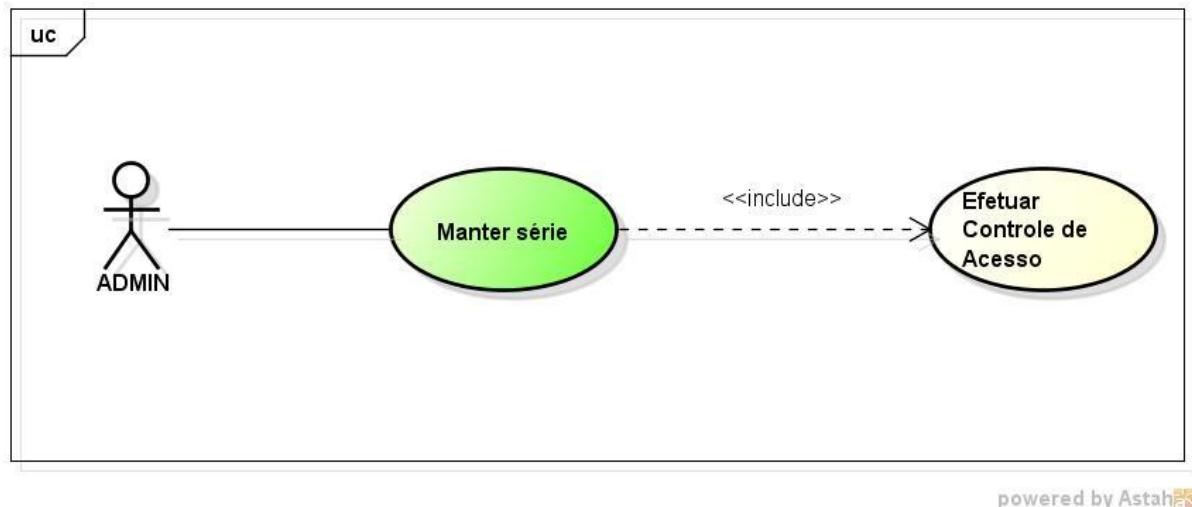


Figura 31- UC - Manter serie

Nome do Caso de Uso	Manter série
Atores	Administrador
Pré-Condições	Efetuar Controle de Acesso
Cenário Principal	1- Administrador informa a série e em seguida seleciona confirmar. 2- O sistema carrega os dados. 3- O Administrador finaliza a operação.
Cenário Alternativo	Administrador pode cancelar a operação durante o processo.
Casos de Teste	1- Os campos obrigatórios estando preenchidos o sistema executa a operação. 2- Caso contrário, o sistema retorna uma mensagem informando quais campos devem ser informados.

Tabela 26- Manter serie

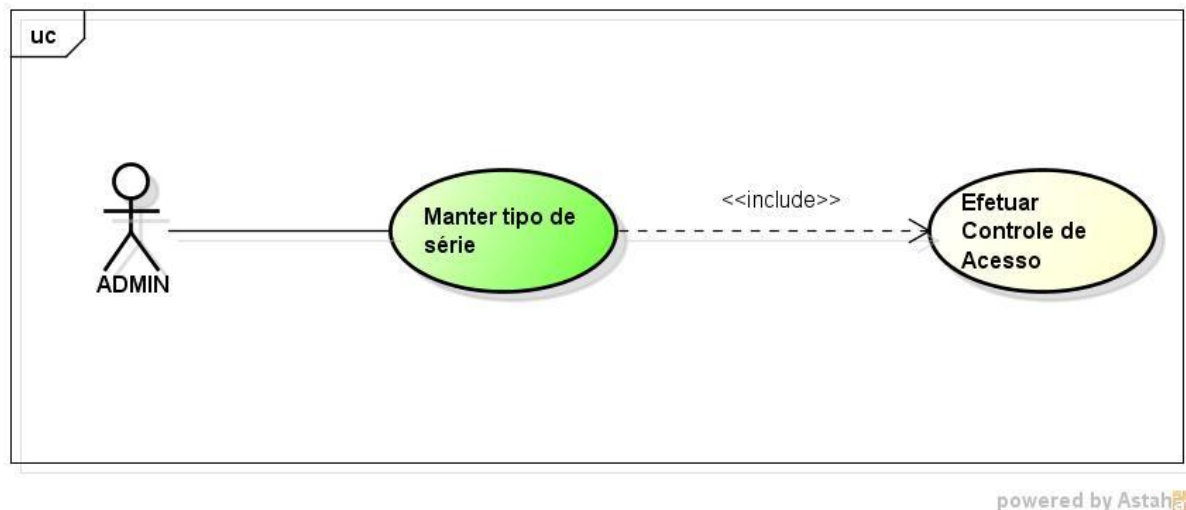


Figura 32- UC - Manter tipo de série

Nome do Caso de Uso	Manter tipo de série
Atores	Administrador
Pré-Condições	Efetuar Controle de Acesso
Cenário Principal	1- Administrador informa o tipo de série e em seguida seleciona confirmar. 2- O sistema carrega os dados. 3- O Administrador finaliza a operação.
Cenário Alternativo	Administrador pode cancelar a operação durante o processo.
Casos de Teste	1- Os campos obrigatórios estando preenchidos o sistema executa a operação. 2- Caso contrário, o sistema retorna uma mensagem informando quais campos devem ser informados.

Tabela 27- Manter tipo de série

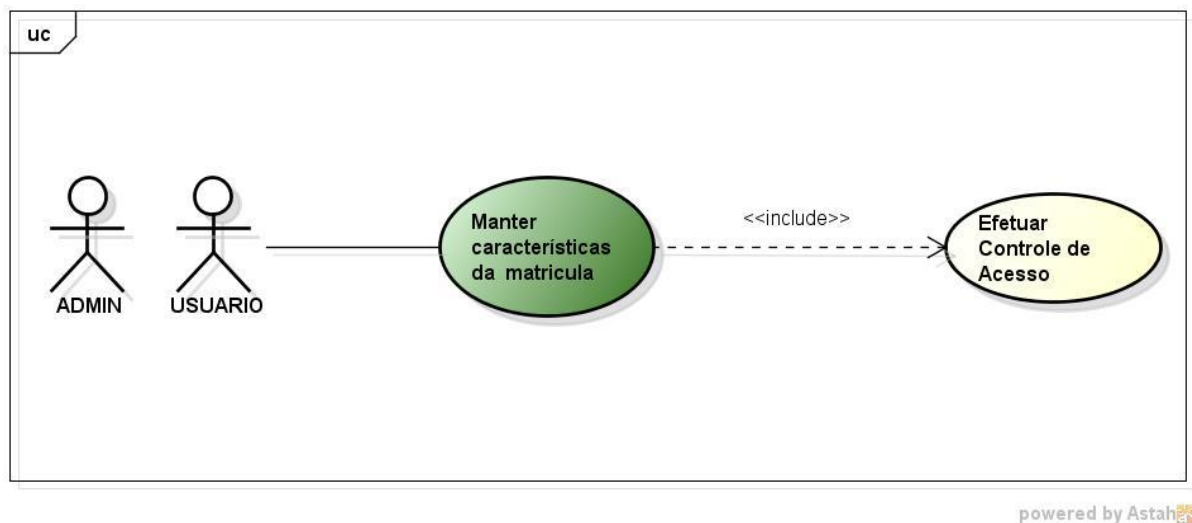


Figura 33- UC - Manter características da matrícula

Nome do Caso de Uso	Manter características da matrícula
Atores	Administrador/Usuário
Pré-Condições	Efetuar Controle de Acesso
Cenário Principal	1- Administrador/Usuário informa as características da matrícula e em seguida seleciona confirmar. 2- O sistema carrega os dados. 3- O Administrador/Usuário finaliza a operação.
Cenário Alternativo	Administrador/Usuário pode cancelar a operação durante o processo.
Casos de Teste	1- Os campos obrigatórios estando preenchidos o sistema executa a operação. 2- Caso contrário, o sistema retorna uma mensagem informando quais campos devem ser informados.

Tabela 28- Manter características da matrícula

3.1.4 Diagramas de atividades

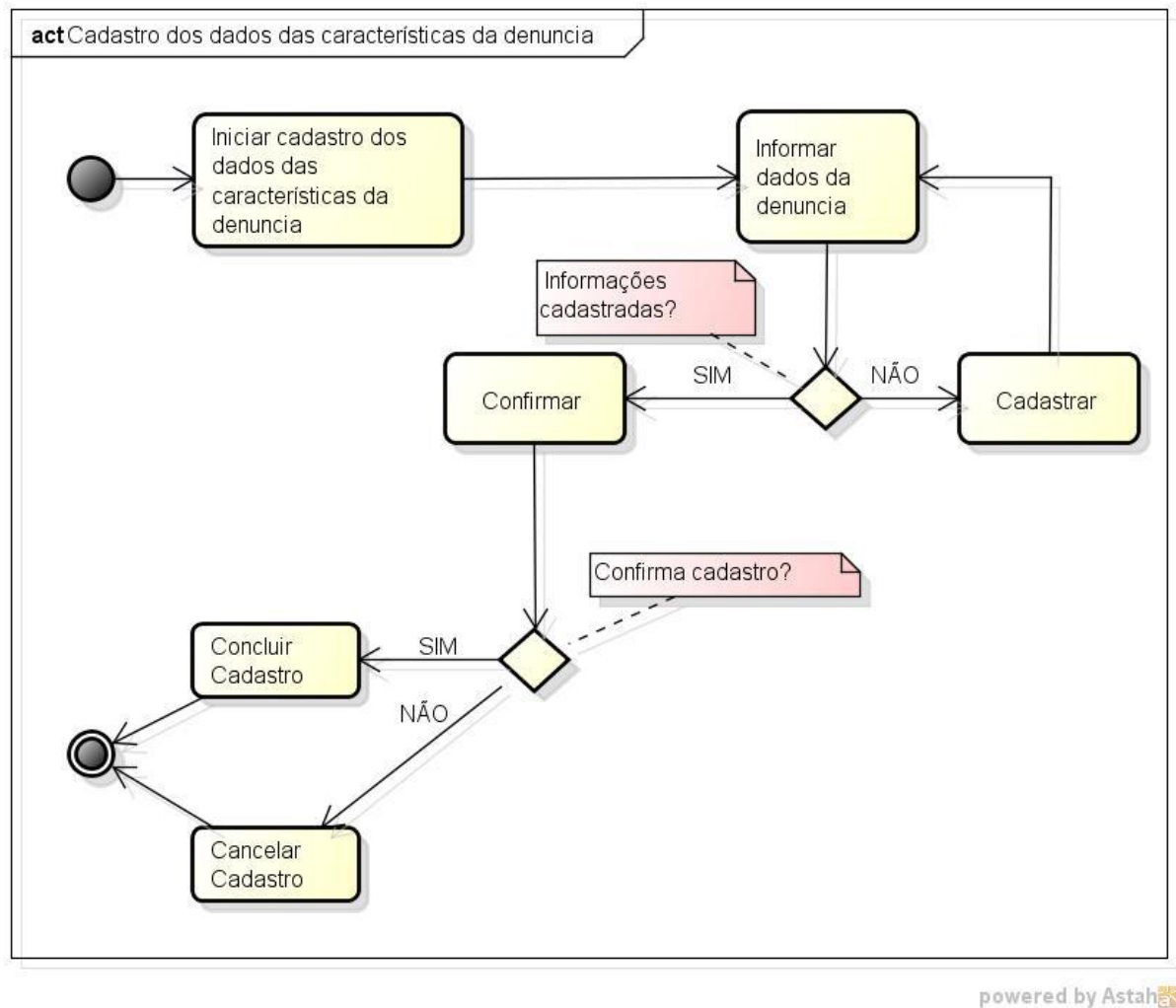


Figura 34- Diagrama de Atividade - Cadastro das características da denuncia

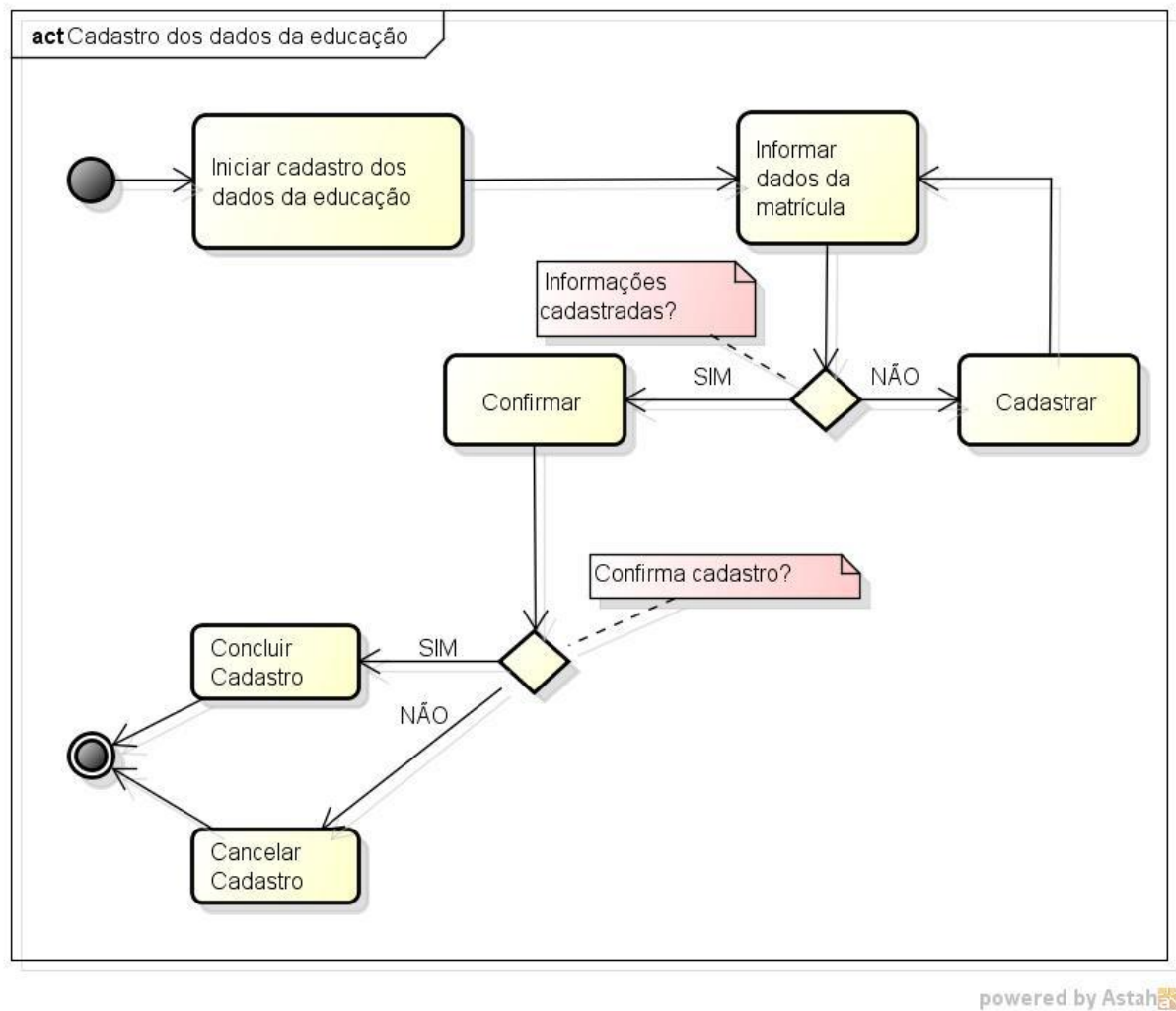


Figura 35- Diagrama de Atividade - Cadastro dos dados da educação

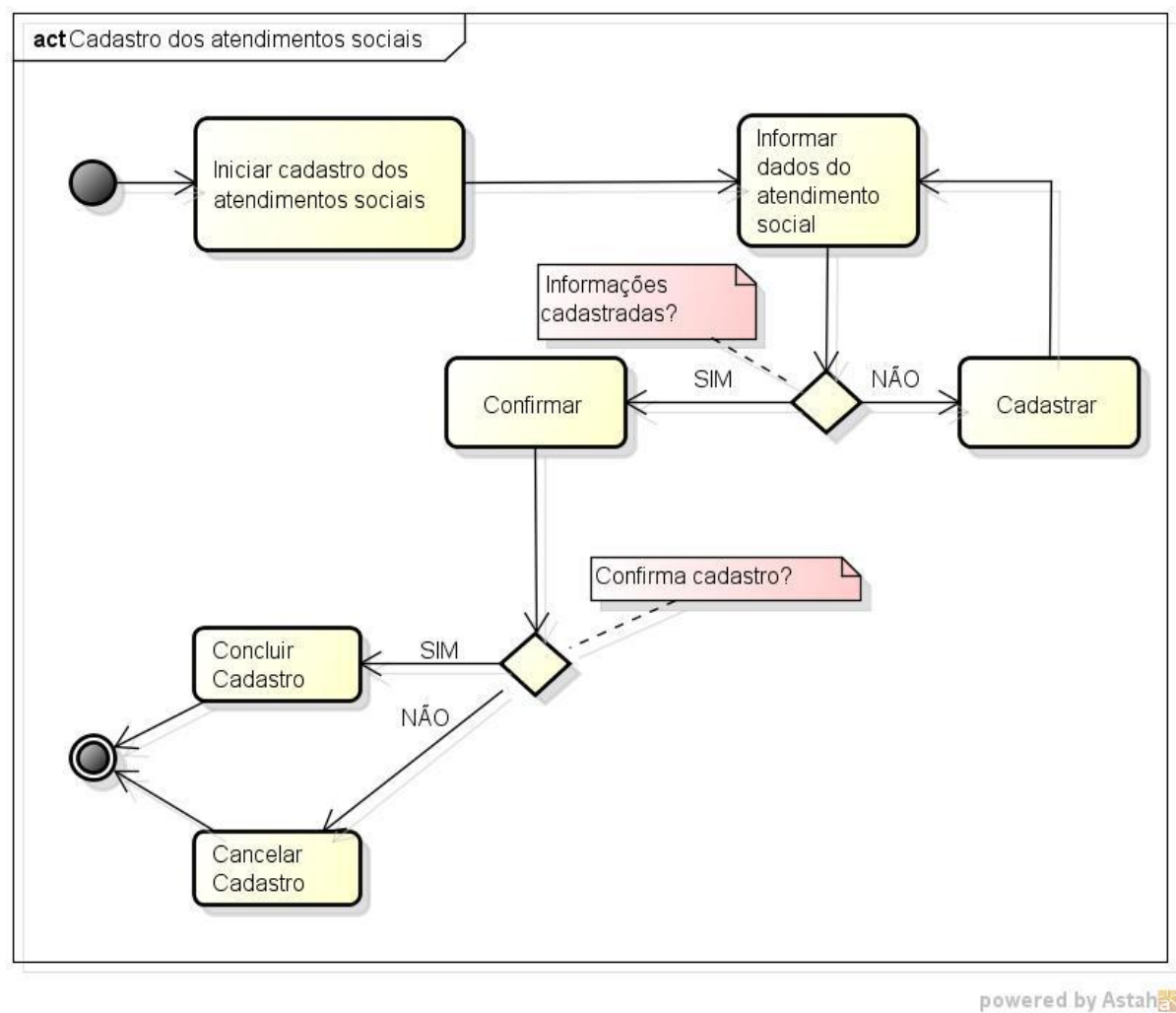


Figura 36- Diagrama de Atividades - Cadastro das informações de atendimentos sociais

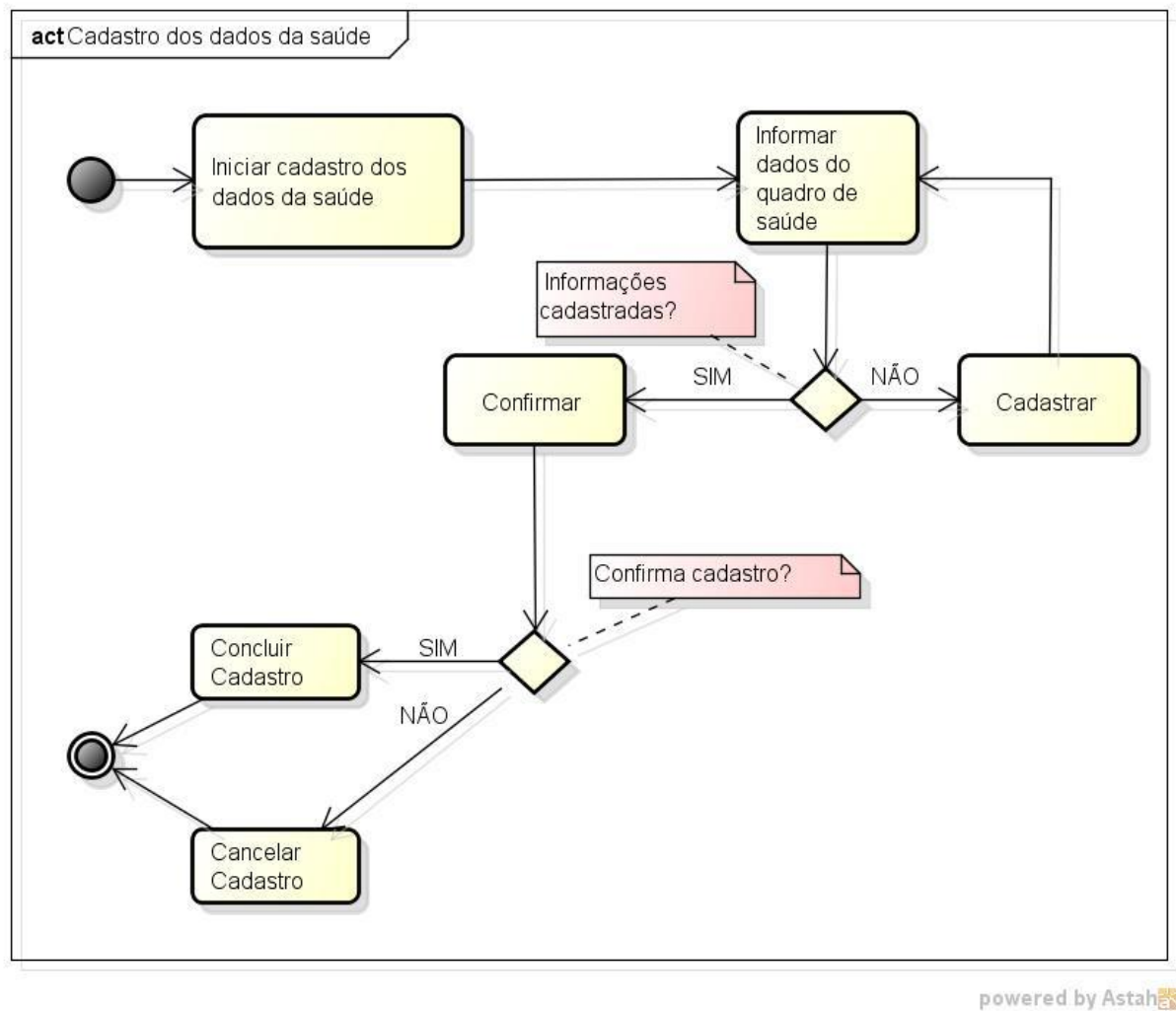


Figura 37- Diagrama de Atividades - Cadastro dos dados da saúde

3.1.5 Diagramas de sequência

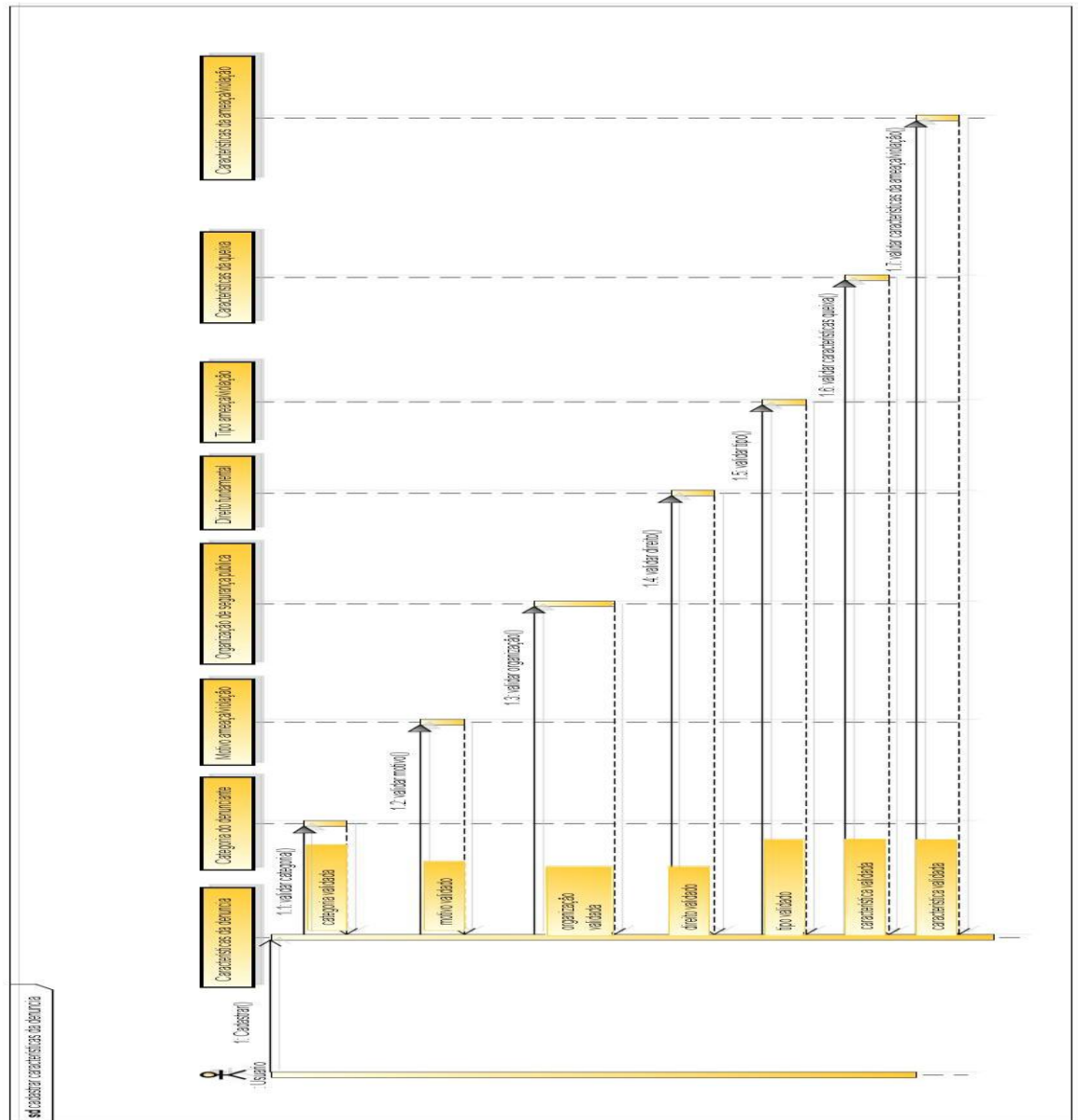


Figura 38- Diagrama de sequência - cadastrar característica da denuncia

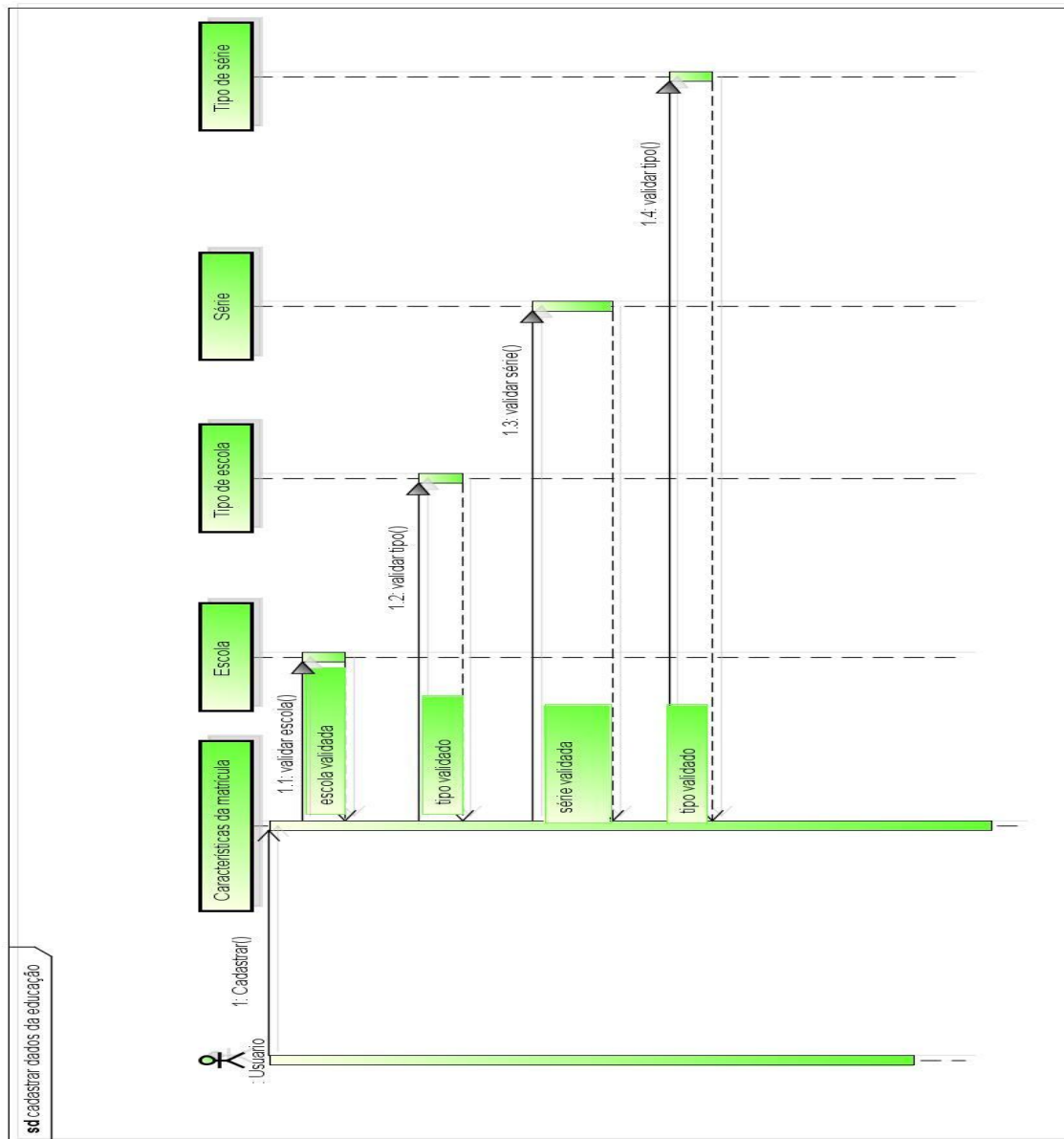


Figura 39- Diagrama de sequência - cadastrar os dados da educação

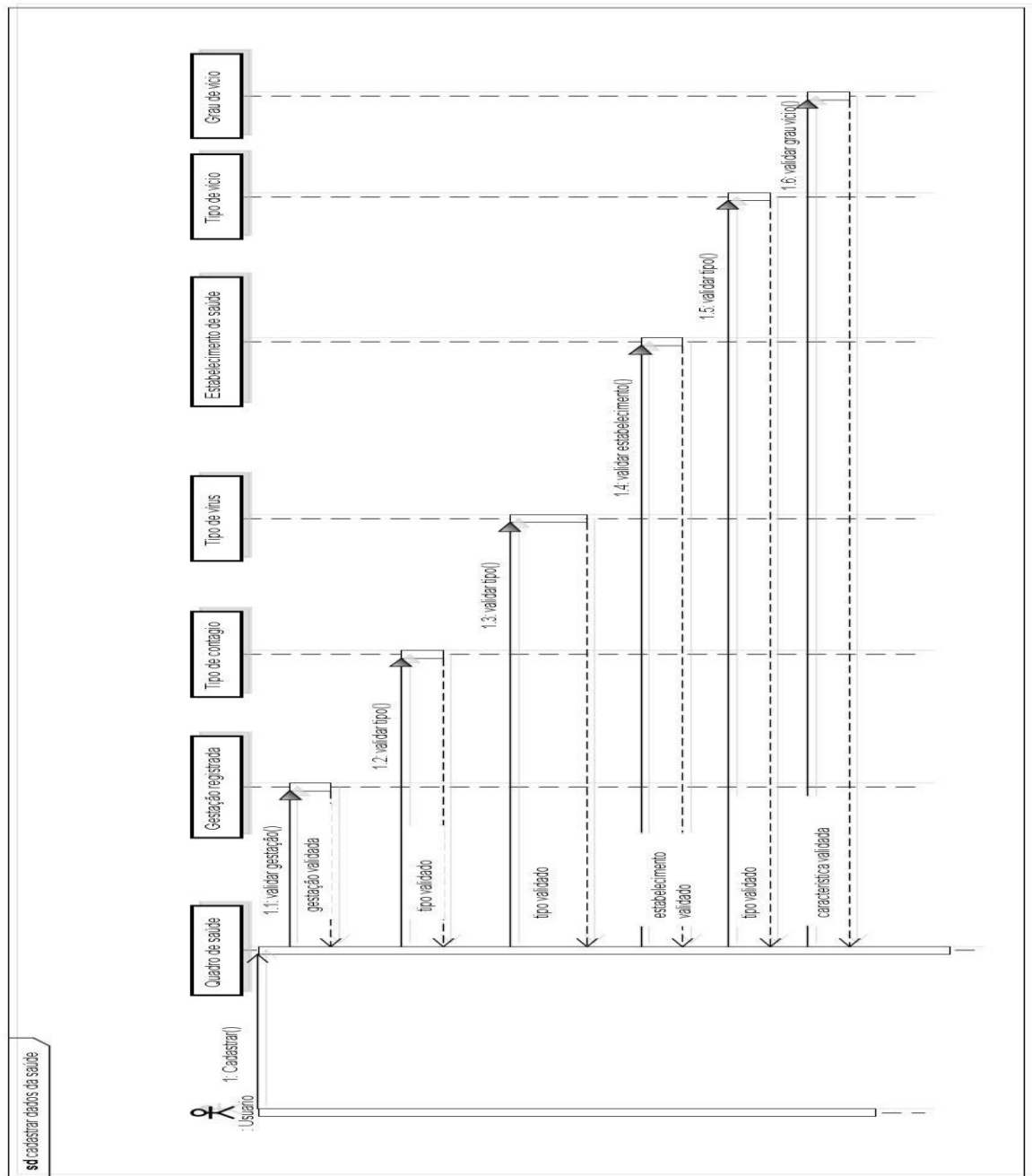


Figura 40- Diagrama de sequência - cadastrar quadro de saúde

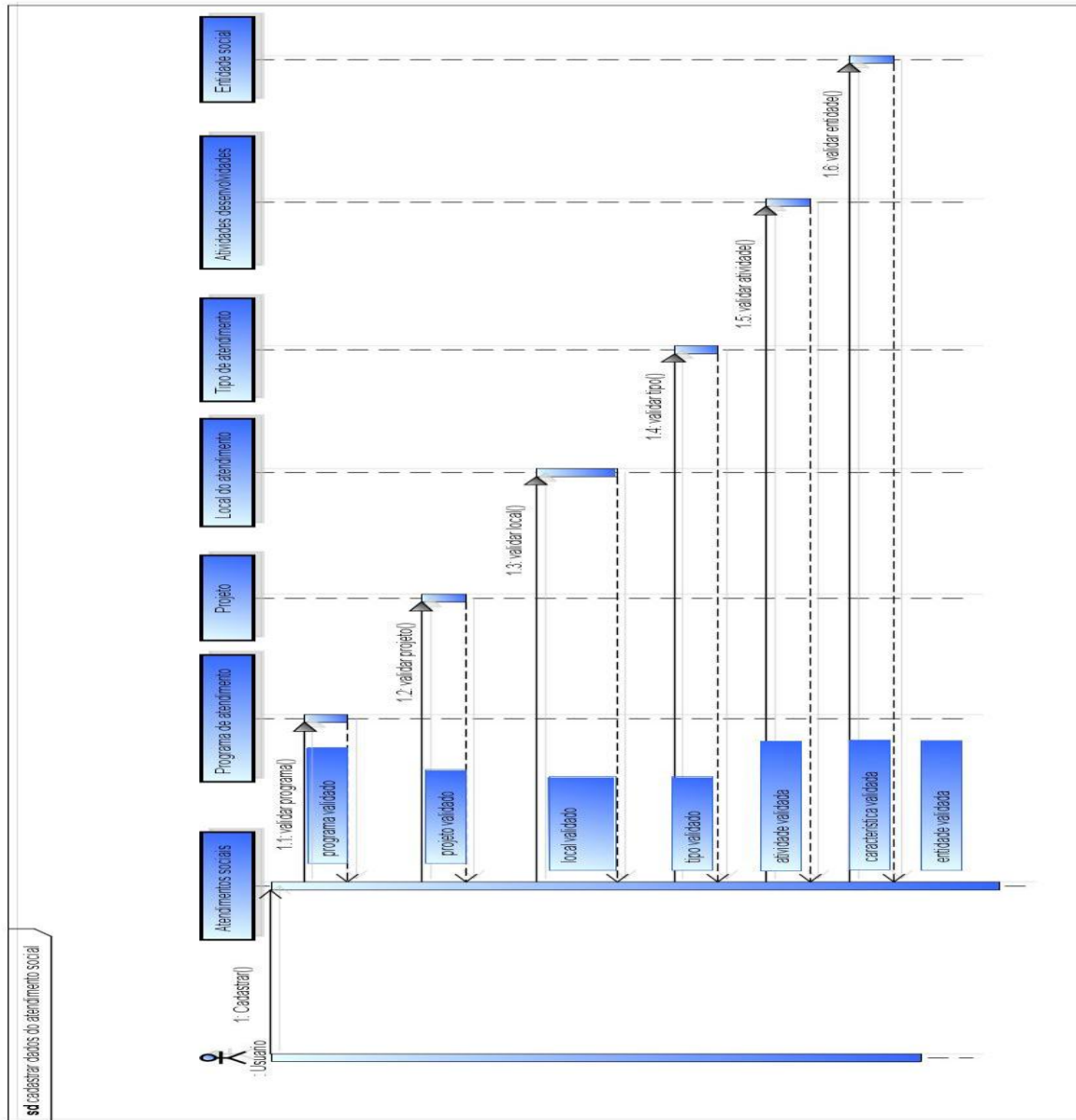


Figura 41- Diagrama de sequência - cadastrar atendimento social

3.1.6 Diagrama de classe

O diagrama abaixo representa as classes dos setores de segurança pública, educação, saúde e atendimento social.

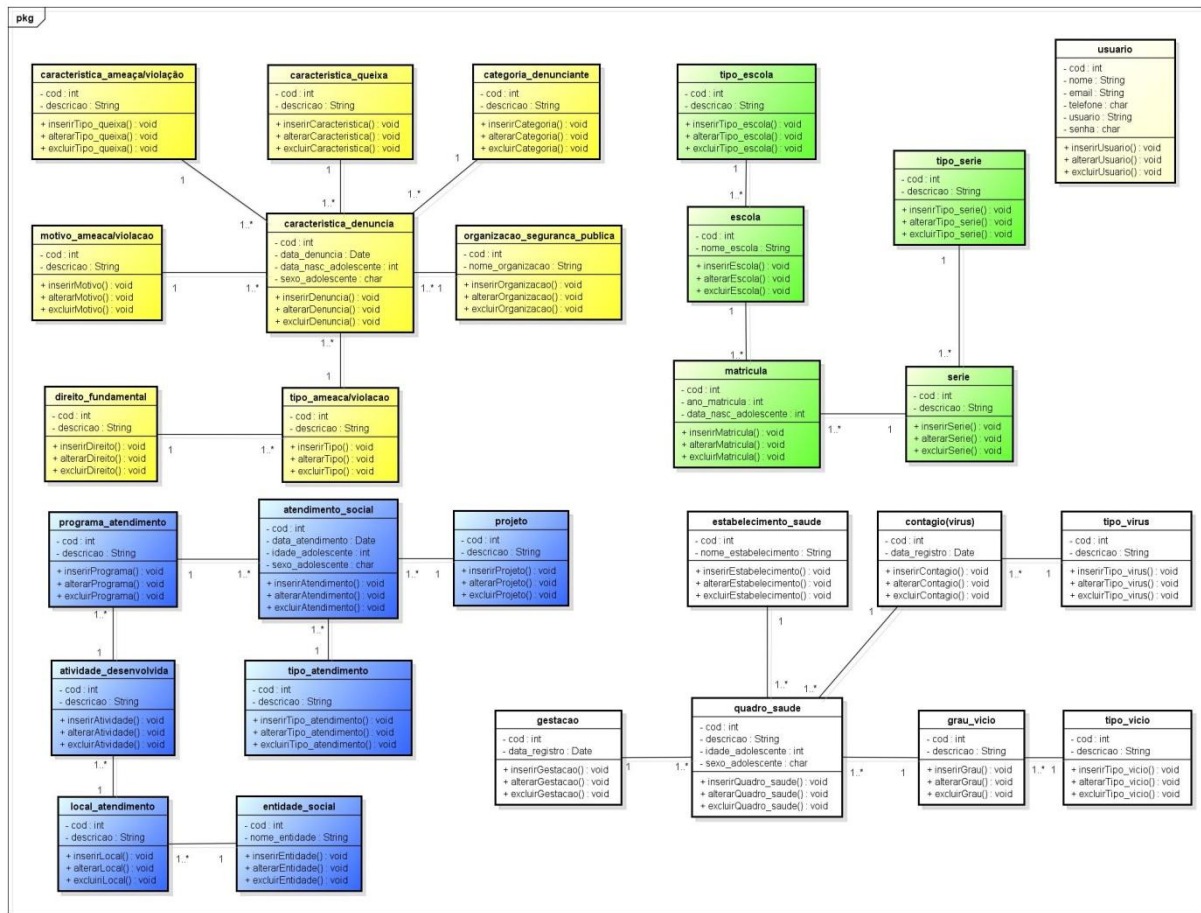


Figura 42- Diagrama de classe

- Amarelo – Segurança Pública: Categoria do denunciante, Motivo da ameaça ou violação, Organização de segurança pública, Direito fundamental, Tipo da ameaça ou violação, Características de queixa, Características da ameaça ou violação, Características da denúncia.

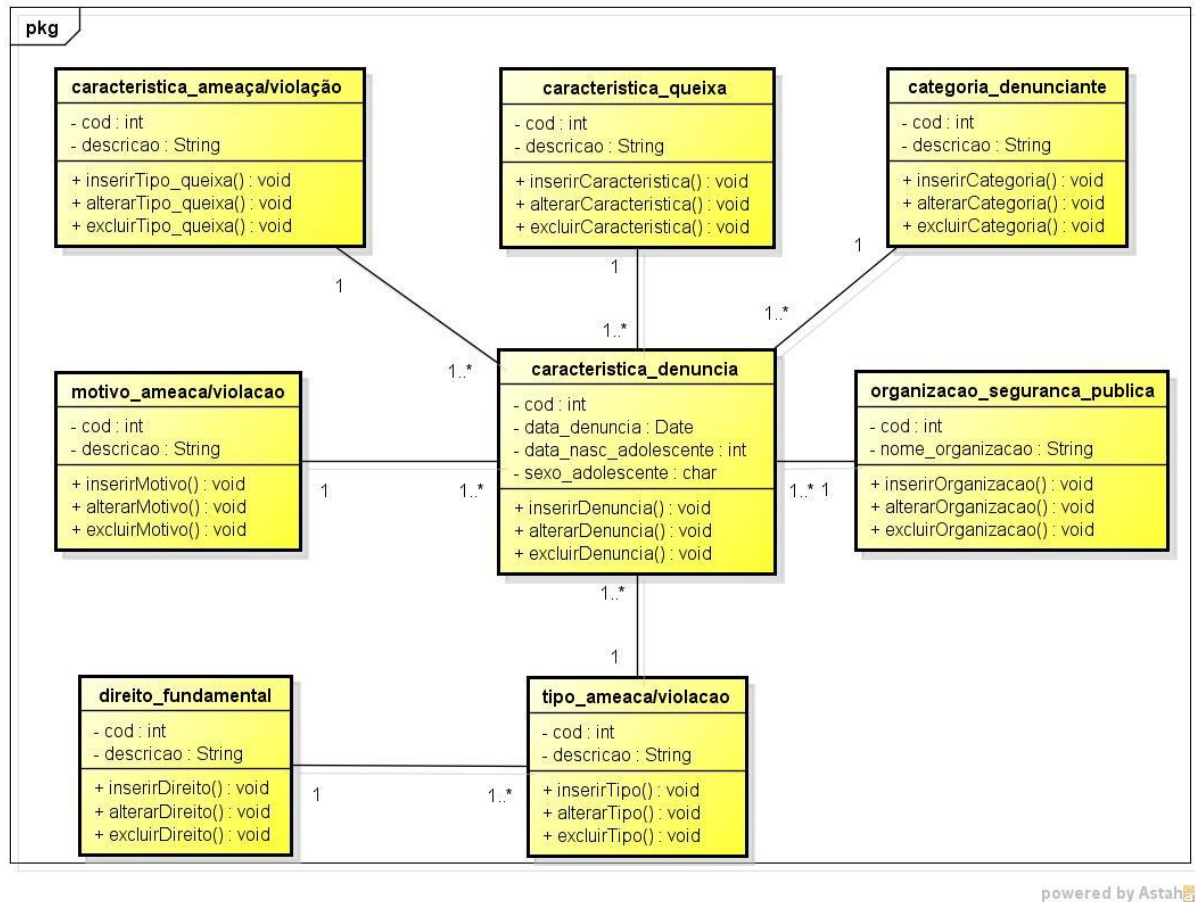
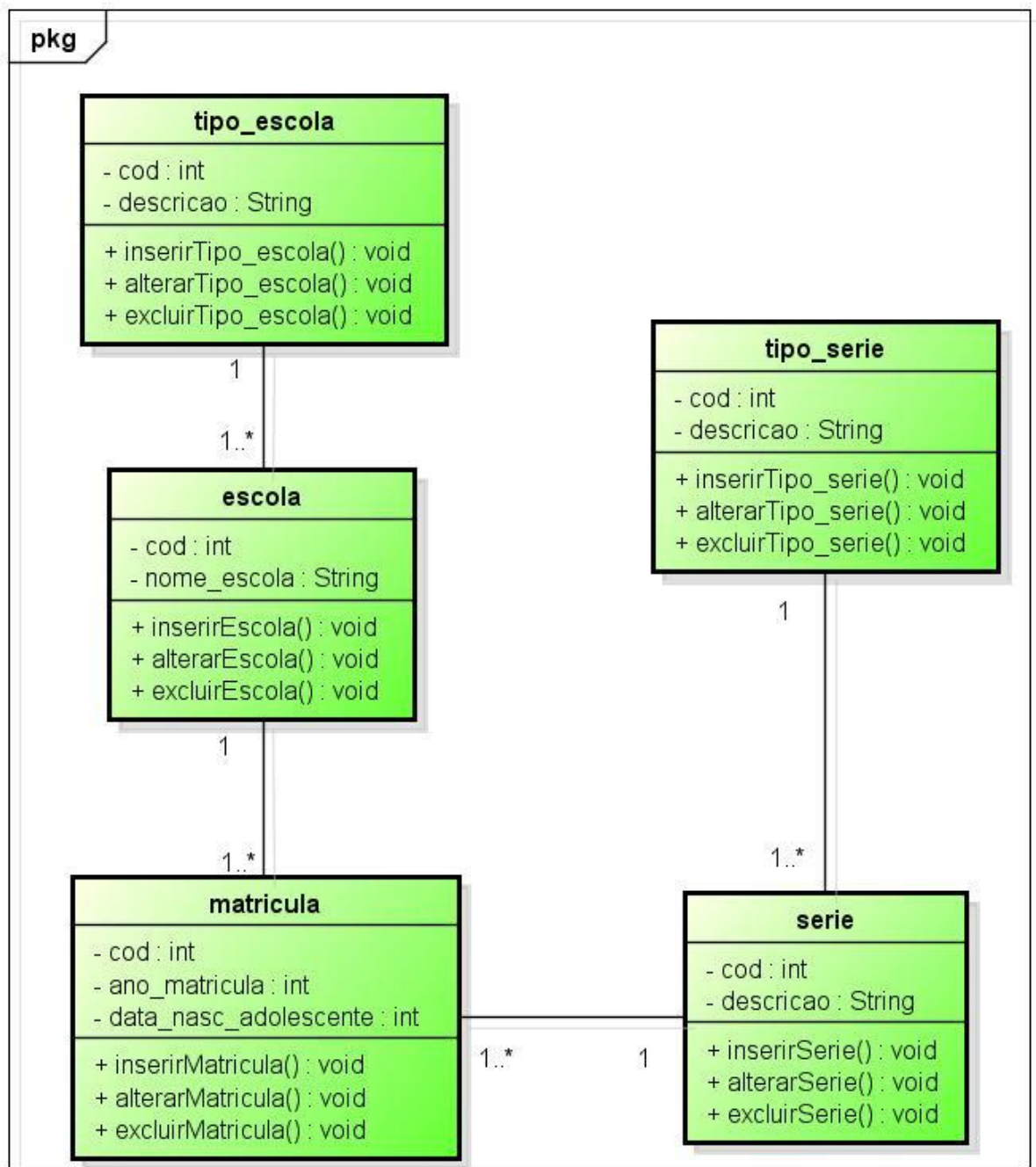


Figura 43- Diagrama de classe - Setor de Segurança Pública

- Verde – Educação: Matrícula, Escola, Tipo da escola, Série, Tipo de série, Dados da matrícula.



powered by Astah

Figura 44- Diagrama de classe - Setor da Educação

- Azul – Atendimento Social: Programa de atendimento, Projeto, Local do atendimento, Tipo de atendimento, Atividades desenvolvidas, Entidades sociais, Características do atendimento.

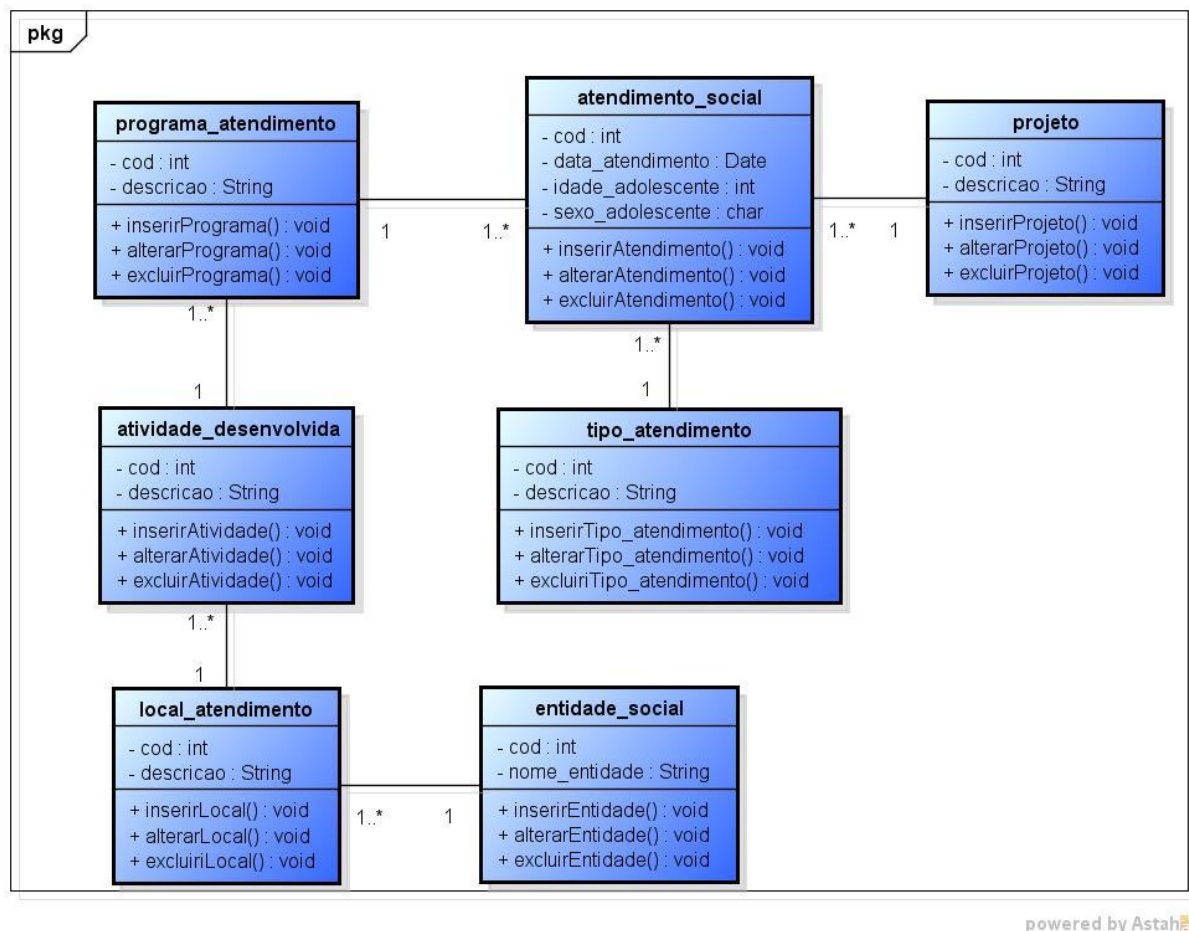


Figura 45- Diagrama de classe - Atendimentos Sociais

- Branco – Saúde: Quadro de saúde, gestação, contágios de vírus, estabelecimento de saúde e uso de drogas;

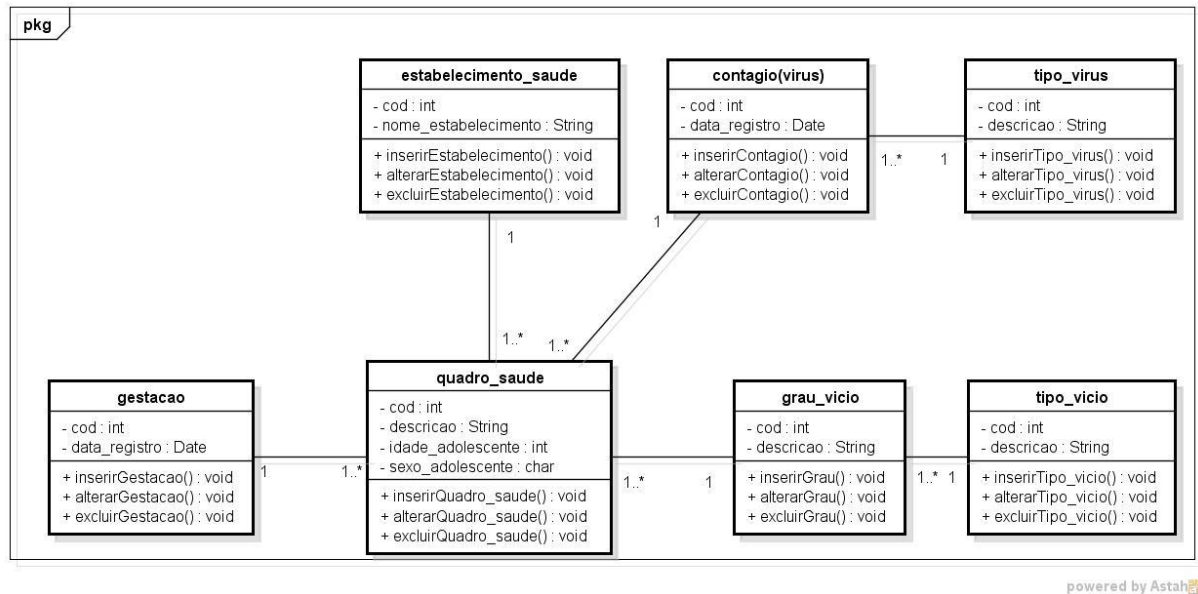


Figura 46- Diagrama de classe - Setor da saúde

3.1.7 Diagrama entidade e relacionamento (DER)

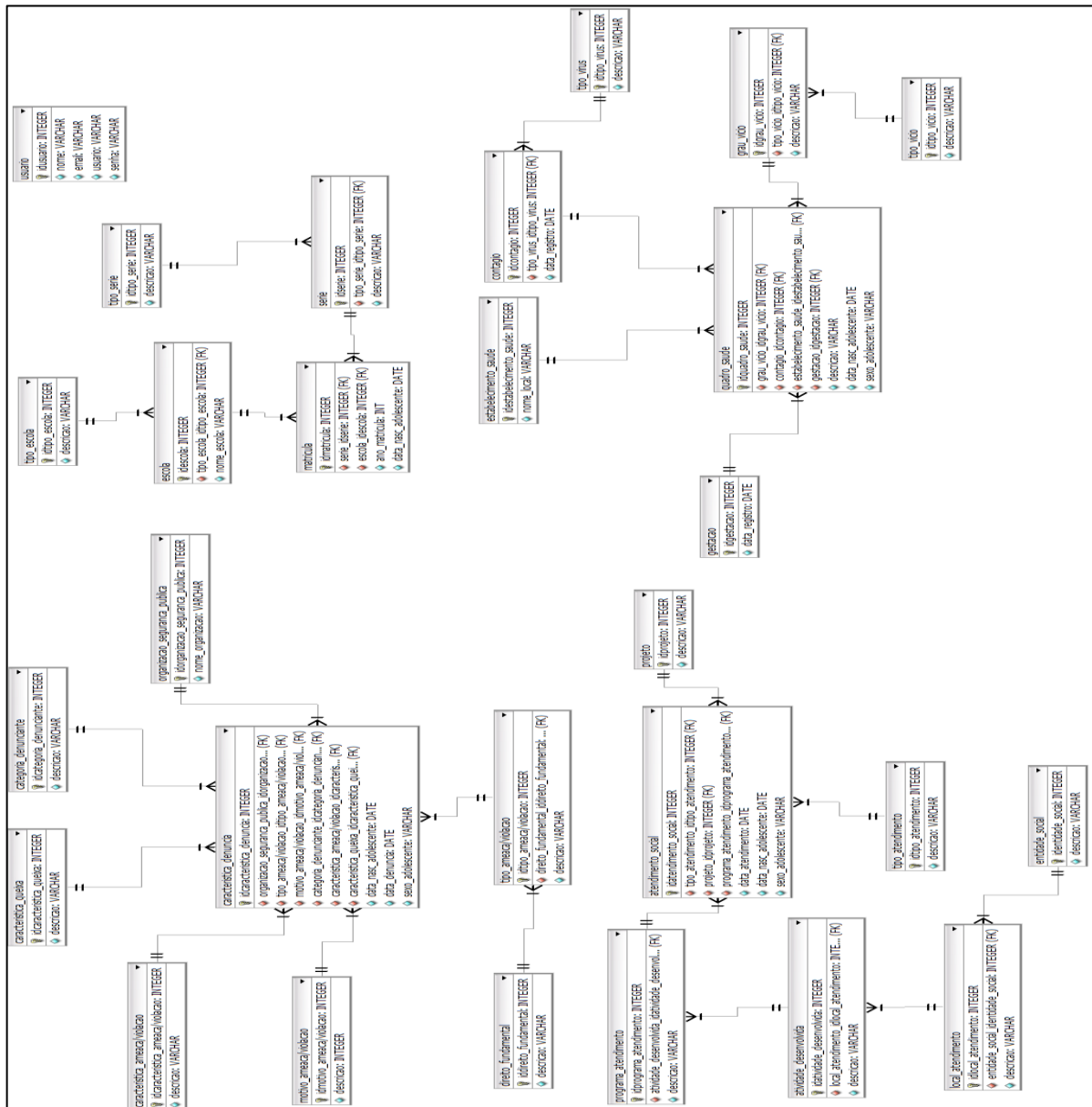


Figura 47- Diagrama Entidade Relacionamento

3.2 PLANEJAMENTO DO PROJETO

3.2.1 Estrutura analítica de trabalho (WBS)

WBS é uma estrutura de subdivisão do projeto, organiza e define cada passo do projeto. O trabalho planejado é contido dentro dos componentes de nível mais baixo da WBS, que são chamados de pacotes de trabalho. O pacote de trabalho pode ser agendado, controlado e monitorado. Após a criação da WBS no projeto; o planejamento do projeto está todo orientando e estruturado para a equipe de execução do projeto, bem como para as demais partes interessadas. (<http://www.tiespecialistas.com.br>).

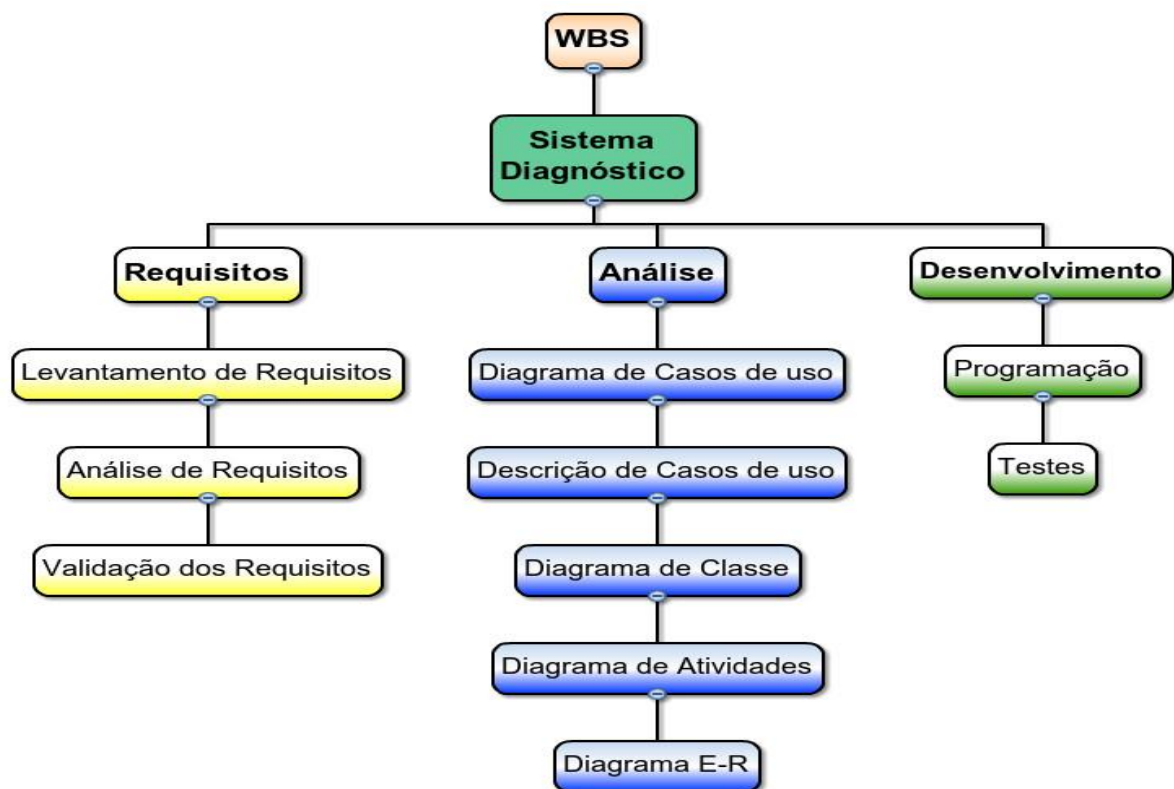


Figura 48- WBS

3.2.2 Orçamento

Os recursos necessários para a elaboração da análise e do desenvolvimento do sistema de diagnóstico são:

- 01 Analista de Sistema
- 01 Programador
- 01 Notebook

Analista de Sistemas			
Analista	Quantidade de dias	Custo dia	Total
Tatiane Schwartz Borges	150	R\$ 35,00	R\$ 5.250,00
Programador			
Programador	Quantidade de dias	Valor Hora	Total
Tatiane Schwartz Borges	120	R\$ 30,00	R\$ 3.600,00
			Total R\$ 8.850,00

Figura 49 - Orçamento - Analista/Programador

Equipamentos			
Equipamento	Valor	Depreciação (2 anos)	Total
Notebook	R\$ 2.600,00	R\$ 4,16	R\$ 915,20
			Total R\$ 915,20

Figura 50 - Orçamento - Equipamentos

TOTAL	
Mão de obra	R\$ 8.850,00
Equipamentos	R\$ 915,20
TOTAL R\$ 9.765,20	

Figura 51 - Orçamento - Valor Total

3.2.3 Cronograma de atividades

Cronograma das atividades que serão desenvolvidas para o trabalho de conclusão do curso:

Atividades/Período	2013			2014								
	Out	Nov	Dez	Jan	Fev	Mar	Abr	Mai	Jun	Jul	Ago	Set
✓ Leitura e estudos de referências bibliográficas	X	X	X	X	X	X	X	X	X	X		
✓ Elaboração do pré-projeto	X	X										
✓ Entrega do pré-projeto		X										
✓ Estudos sobre a elaboração do diagnóstico		X	X	X	X							
✓ Estudos sobre as ferramentas que serão utilizadas		X	X	X	X							
✓ Levantamento de requisitos		X	X	X	X	X						
✓ Preparação para a qualificação		X	X	X	X	X						
✓ Entrega e apresentação da qualificação						X	X					
✓ Implementação do sistema							X	X	X	X		
✓ Entrega e apresentação do trabalho final											X	X

Figura 52- Cronograma de Atividades

- ✓ Atividades concluídas.

4 - DESENVOLVIMENTO DO SISTEMA

Para a implementação do Sistema de Diagnóstico foi utilizado o ambiente de desenvolvimento Eclipse com a linguagem de programação Java e o framework de persistência Hibernate. Além do banco de dados MySQL, responsável por armazenar os dados.

4.1 ORGANIZAÇÃO DO PROJETO

Para uma melhor organização do sistema, o projeto foi dividido em pacotes. O projeto está dividido em três pacotes são eles: bean, dao, modelo conforme ilustra a figura 53.

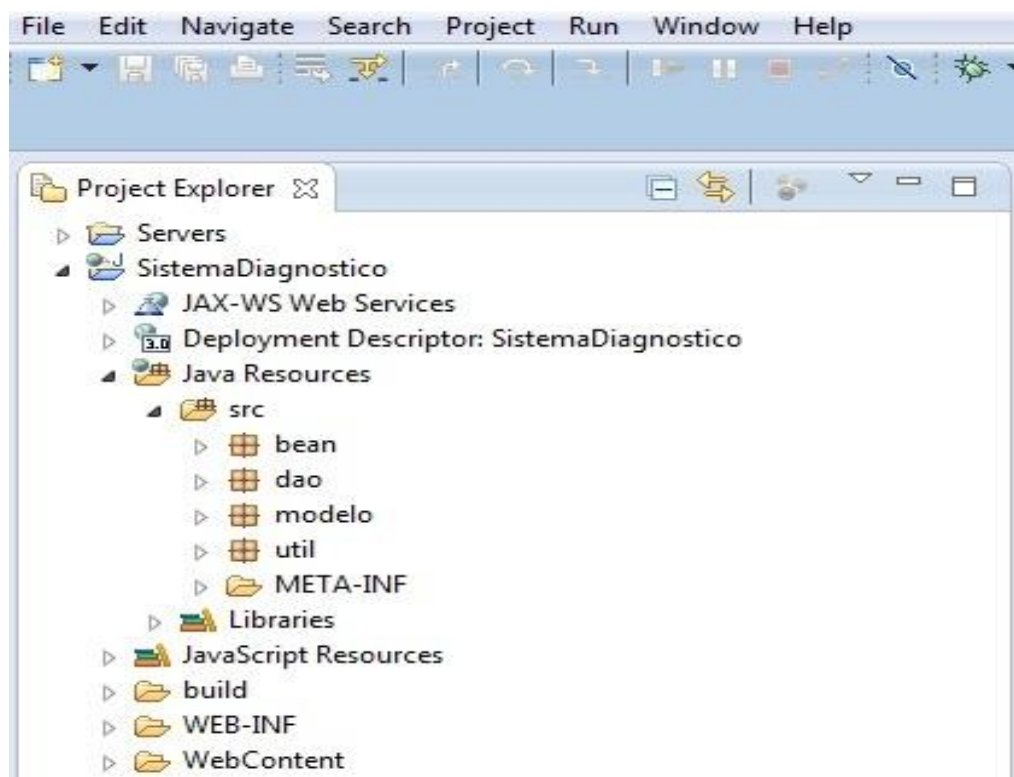


Figura 53 - Organização dos pacotes

4.1.1 Pacote modelo

O pacote modelo é onde se encontra as classes de modelo, que possuem os métodos construtores e de acesso a valores de atributos da classe, do tipo get e set, além dos respectivos mapeamentos da camada de persistência por meio de anotações Java. A figura 54 representa os mapeamentos da classe.

```
package modelo;

import java.io.Serializable;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.Table;

@Entity
@Table
public class CategoriaDenunciante implements Serializable {

    private static final long serialVersionUID = -2085898227295600873L;

    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    @Column(name = "idCategoria")
    private int id;

    @Column(name = "descricao", length = 200)
    private String descricao;

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public String getDescricao() {
        return descricao;
    }

    public void setDescricao(String descricao) {
        this.descricao = descricao;
    }

}
```

Figura 54 - Mapeamentos classe modelo1

Para o mapeamento das classes são utilizadas as anotações do Hibernate. Qualquer símbolo em um código Java que comece com uma @ (arroba) é uma anotação.

Todas as classes persistentes mapeadas com anotações são declaradas usando a anotação **@Entity**. A anotação **@Table** indica qual é o nome da tabela mapeada, usando o atributo **name**. Veja na Figura 55 a utilização destes conceitos.

A chave primária da tabela é mapeada na classe através da anotação **@Id**. A anotação **@GeneratedValue** permite a definição automática para o valor da chave primária e a anotação **@Column** informando o nome da coluna, através do atributo **name**. (<http://www.futurepages.org/>)

```
package modelo;

import java.io.Serializable;[...]

@Entity
@Table (name="CaracteristicaDenuncia")
public class CaracteristicaDenuncia implements Serializable {

    private static final long serialVersionUID = -8575047442435281733L;

    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    @Column(name = "idCaracteristica")
    private int id;
    @Column(name = "dataDenuncia", length = 10)
    private Date dataDenuncia;
    @Column(name = "idade", length = 2)
    private int idade;
    @Column(name = "sexo", length = 10)
    private String sexo;
    @Column(name = "bairro", length = 200)
    private String bairro;

    @ManyToOne (fetch = FetchType.EAGER)
    @JoinColumn(name="idCategoria", insertable=true, updatable=true)
    @Fetch(FetchMode.JOIN)
    @Cascade(CascadeType.SAVE_UPDATE)

    private CategoriaDenunciante categoriaDenunciante = new
    CategoriaDenunciante();
```

Figura 55 - Mapeamentos classe modelo2

A anotação **@ManyToOne** representa o relacionamento muitos para um (n-1) entre as tabelas. Esta anotação também possui o atributo **@fetch**, que define como o atributo mapeado será recuperado da base de dados.

A anotação **@JoinColumn** é utilizada para informar qual o nome da coluna que corresponde à chave estrangeira do mapeamento. Nesta anotação, também são acrescentados os atributos **insertable** e **updatable** que se assumirem o valor true, indica que o objeto deve ser inserido ou atualizado no momento em que o objeto que possui o relacionamento é inserido ou atualizado.

Outra anotação é a **@Cascade**, que serve para indicar com que ação em cascata o relacionamento será tratado, ou seja, quais operações deverão ser em cascata do objeto pai para o objeto associado. (<http://www.futurepages.org/>)

4.1.2 Pacote dao

O pacote dao é onde se localiza as classes que vão gerenciar os dados, fazendo a comunicação com o banco de dados. A Figura 56 descreve os métodos que são utilizados.

```

package dao;
import java.util.List;
public class CaracteristicaDenunciaDAO {

    private EntityManager entityManager;

    public CaracteristicaDenunciaDAO() {
        entityManager = EntityFactory.getEntityManager();
    }

    public CaracteristicaDenuncia getById(final int id) {
        return entityManager.find(CaracteristicaDenuncia.class, id);
    }

    @SuppressWarnings("unchecked")
    public List<CaracteristicaDenuncia> findAll() {
        return entityManager.createQuery("FROM " +
CaracteristicaDenuncia.class.getName()).getResultList();
    }

    public void persist(CaracteristicaDenuncia CaracteristicaDenuncia) {
        try {
            entityManager.getTransaction().begin();
            entityManager.persist(CaracteristicaDenuncia);
            entityManager.getTransaction().commit();
        } catch (Exception ex) {
            ex.printStackTrace();
            entityManager.getTransaction().rollback();
        }
    }
}

```

Figura 56 - Classe dao - Métodos persist/List

O método List (Figura 56) retorna uma lista de CaracterísticaDenuncia através da query criada. O método persist é responsável pela inclusão dos dados, a imagem acima ilustra a inclusão das características da denuncia.

O método receberá um objeto do tipo CaracterísticaDenuncia, em seguida abrirá uma transação através do `getTransaction().begin()` e persistirá o objeto recebido com `persist(CaracteristicaDenuncia)`. Após a persistência a transação será encerrada com `getTransaction().commit()`. E para alteração usasse o método merge e para a exclusão dos dados usasse o método remove, como ilustra a figura 57.

```

public void merge(CaracteristicaDenuncia CaracteristicaDenuncia) {
    try {
        entityManager.getTransaction().begin();
        entityManager.merge(CaracteristicaDenuncia);
        entityManager.getTransaction().commit();
    } catch (Exception ex) {
        ex.printStackTrace();
        entityManager.getTransaction().rollback();
    }
}

public void remove(CaracteristicaDenuncia CaracteristicaDenuncia) {
    try {
        entityManager.getTransaction().begin();
        entityManager.remove(CaracteristicaDenuncia);
        entityManager.getTransaction().commit();
    } catch (Exception ex) {
        ex.printStackTrace();
        entityManager.getTransaction().rollback();
    }
}

public void removeById(final Integer id) {
    try {
        CaracteristicaDenuncia caracteristica = getById(id);
        remove(caracteristica);
    } catch (Exception ex) {
        ex.printStackTrace();
    }
}
}

```

Figura 57 - Classe dao - Métodos merge/remove

4.1.3 Pacote bean

O pacote bean representa a comunicação da camada de visão representada pelas páginas JSF com o banco de dados sendo responsável por realizar as operações requisitadas nas páginas de interação com o usuário, como ilustra a figura 58.

```

package bean;

import java.io.Serializable;

@ManagedBean(name = "CategoriaDenuncianteMB")
@ViewScoped
public class CategoriaDenuncianteMB implements Serializable {

    private static final long serialVersionUID = 5187937123528091468L;
    private static CategoriaDenunciante categoria = new
CategoriaDenunciante();
    private List<CategoriaDenunciante> listcategoria = new
ArrayList<CategoriaDenunciante>();
    private int id;

    public CategoriaDenuncianteMB() {
        getListCategoriaDenunciante();
    }
    public String cadastrar() {

        CategoriaDenuncianteDAO ed = new CategoriaDenuncianteDAO();
        ed.persist(categoria);
        categoria = new CategoriaDenunciante();
        getListCategoriaDenunciante();

        return "cadastrocategoriadenunciante.xhtml?faces-redirect=true";
    }
    public String cancelarAlterar(){
        categoria = new CategoriaDenunciante();
        return "cadastrocategoriadenunciante.xhtml?faces-redirect=true";
    }
    public void excluir(int c) {
        CategoriaDenuncianteDAO ed = new CategoriaDenuncianteDAO();
        ed.removeById(c);
        categoria = new CategoriaDenunciante();
        getListCategoriaDenunciante();
    }
    public String Alterar() {
        CategoriaDenuncianteDAO ed = new CategoriaDenuncianteDAO();
        ed.merge(categoria);
        categoria = new CategoriaDenunciante();
        getListCategoriaDenunciante();

        return "cadastrocategoriadenunciante.xhtml?faces-redirect=true";
    }

    public String Carregarpagina(CategoriaDenunciante c) {

        categoria = c;
        return "alterarcadastrocategoriadenunciante.xhtml?faces-
redirect=true";
    }
}

```

Figura 58 - Classe bean

Além dos pacotes de codificação de classes, também há alguns diretórios importantes:

- **Src:** A pasta src é onde se encontrasse todos os códigos fonte Java (.java).
- **META-INF:** Onde se localiza o arquivo persistence.xml. O arquivo persistence.xml é responsável por diversas configurações do sistema.
- **Build:** Onde o Eclipse compila as classes (.class).
- **WEB-INF:** Onde se localiza o arquivo web.xml. Este arquivo é responsável por configurar as páginas web.
- **WebContent:** Dentro da pasta WebContent, é onde se localiza as páginas, imagens, css, etc.
- **WebContent/WEB-INF/lib/ :** Onde se localiza as bibliotecas necessárias do sistema.

4.2 INTERFACES DO SISTEMA

Ao acessar o sistema de diagnóstico o usuário/administrador do sistema, deve preencher os campos obrigatórios de usuário e senha. Depois de validados, o usuário terá acesso ao sistema, abrirá a página inicial contendo um menu com os campos opcionais de cadastros, são eles: Atendimentos sociais, educação, saúde, violações de direitos e cadastros de usuários.

Ao selecionar uma das opções de menus, abrirá um submenu onde se localiza os campos cadastros, gerar cadastro e relatórios. Se o sistema for acessado por um usuário comum, o mesmo só terá permissões para gerar cadastro e visualizar relatórios, caso o usuário seja um administrador o mesmo terá o total acesso, podendo administrar o sistema.



Figura 59 - Tela de controle de acesso

Se o acesso ao sistema for efetuado por um administrador, será direcionado para a tela de gerenciamento do sistema, onde poderão ser gerenciadas todas as atividades da aplicação. A figura 60, ilustra a interface inicial do sistema.



Figura 60 - Tela Menu do Administrador

O administrador poderá efetuar os cadastros, excluir e fazer as alterações necessárias nos registros, conforme ilustra as figuras 61 e 62.

Cadastro da categoria do denunciante

Descrição:

Código	Descrição	Ações
4	Família	
5	Sociedade	
6	Entidades	
7	Acompanhamento sem denuncia	

Rede Criança
CONSELHO MUNICIPAL DOS DIREITOS DA CRIANÇA E DO ADOLESCENTE DE ASSIS

Figura 61 - Tela de Pré-cadastro

Alterar a categoria do denunciante

Código:

Descrição:

Rede Criança
CONSELHO MUNICIPAL DOS DIREITOS DA CRIANÇA E DO ADOLESCENTE DE ASSIS

Figura 62 - Tela alterar registro

Se o acesso ao sistema for efetuado por um usuário comum, será direcionado para a tela de menu onde o usuário poderá gerar cadastro e visualizar relatórios. Sendo restrito de algumas atividades do sistema, a figura 63 ilustra este procedimento.



Figura 63 - Tela Menu de usuários

O usuário poderá gerar o cadastro preenchendo os campos obrigatórios e fazer as alterações necessárias nos registros.

Figura 64 - Tela gerar cadastro

Sistema Diagnóstico

localhost:8080/SistemaDiagnostico/alterarcaracteristicadenuncia.xhtml

CONSELHO MUNICIPAL DOS DIREITOS DA CRIANÇA E DO ADOLESCENTE DE ASSIS

Home Atendimentos Violações de direitos Saúde Educação Usuário SAIR

Alterar as características da denuncia

Código: 0

Idade: seleccione a idade

Bairro:

Sexo: ☒ Masculino ☐ Feminino

Data da denuncia:

Tipo Ameaça/Violação: seleccione um tipo de ameaça/violação

Motivo Ameaça/Violação: seleccione um motivo de ameaça/violação

Categoria do Denunciante: seleccione uma categoria do denunciante

Característica de Queixa: seleccione uma característica de queixa

Nome da Organização: seleccione a organização

Caractrística da Ameaça/violação: seleccione a característica

Confirmar Cancelar

Figura 65 - Tela alterar cadastro

5 - CONCLUSÃO

Para o desenvolvimento deste trabalho, inicialmente foi necessário realizar um estudo sobre o diagnóstico municipal da criança e do adolescente, como ele é elaborado, quais as principais informações que são necessárias para a elaboração do relatório, que informa a real situação do público juvenil no município de Assis.

Também foi adquirido conhecimento de como esse diagnóstico é importante para o referido município, pois, por meio de tais informações que relatam a situação dos jovens, é que os poderes públicos, conselhos (CMDCA) e as entidades sociais que atendem crianças e adolescentes, podem intervir, traçar metas e planejar ações, para um futuro melhor.

Com as principais informações que são necessárias para a construção do sistema de diagnóstico, foi realizada a modelagem do sistema, informando detalhadamente sobre o funcionamento do sistema.

A modelagem UML foi de extrema importância para o entendimento do sistema que está sendo desenvolvido, por meio dos diagramas de casos de uso (UC), foi possível identificar as funcionalidades dos usuários. O diagrama de atividades mostrou algumas das rotinas que se destacam no sistema, com o diagrama de sequência foi mostrada a comunicação entre os objetos que compõe o mesmo. Para o entendimento das ligações do sistema, o diagrama de classe foi essencial e para ilustrar o relacionamento entre as entidades do banco de dados, foi então utilizado o diagrama entidade e relacionamento (DER).

Após concluir esta etapa de estudos e modelagem, iniciou-se então o desenvolvimento do sistema de diagnóstico. As tecnologias e ferramentas pesquisadas e que estão sendo utilizadas para a codificação do sistema, mostraram-se eficientes no desenvolvimento do sistema proposto.

Uma das justificativas para a realização do desenvolvimento do sistema, é que hoje a informatização está abrangendo diversas empresas, pois sendo assim com a informatização e o armazenamento dos dados extraídos para o diagnóstico, facilitará o trabalho do conselho CMDCA na realização do mesmo, as organizações poderão

então cadastrar suas informações, e com isso a geração de relatórios servirá para montar o diagnóstico da real situação em dados estatísticos.

O desenvolvimento do projeto proporcionou aprendizado em relação à prática de programação com a linguagem Java, conhecimentos de novas tecnologias e entendimento da importância da análise de projetos.

REFERÊNCIAS BIBLIOGRÁFICAS

DEITEL, Paul et al... **Java como programar**. 8ª Edição. Pearson Prentice Hall, 2010. 1104 p.

GÓES, Wilson Moraes. **Aprenda UML por meio de estudos de caso**. São Paulo: Novatec, 2014. 286 p.

GONÇALVES, Edson. **Dominando JavaServer Faces e Facelets Utilizando Spring 2.5, Hibernate e JPA**. Rio de Janeiro: Editora Ciência Moderna Ltda., 2008. 368 p.

HEUSER, Carlos Alberto. **Projeto de banco de dados**. 6ª Edição. Porto Alegre: Bookman, 2009, 279 p.

LUCKOW, Décio Heinzelmann et al.. **Programação Java para web: aprenda a desenvolver uma aplicação financeira pessoal com as ferramentas mais modernas da plataforma Java**. São Paulo: Novatec, 2010. 631 p.

MACEDO, Alexandre. **Relatórios em Java – JasperReports e iReport**. Disponível em: < <http://www.k19.com.br/>>. Acesso em: 19 de fevereiro de 2014.

RIBAS JUNIOR, Fábio et al... **Conhecer para transformar: Guia para diagnóstico e formulação da política municipal de proteção integral das crianças e adolescentes**. São Paulo: Fundação Telefônica, 2011. 332 p.

RUGGIERI, Ruggero. **WBS – Uma ferramenta importante para o Gerente de Projetos**. Disponível em: <<http://www.tiespecialistas.com.br>>. Acesso em: 19 de fevereiro de 2014.

SOFTONIC Software e Apps, Disponível em: <<http://mysql.softonic.com.br/>>. Acesso em: 11 de fevereiro de 2014.

XAVIER, Ana Lúcia Pintar et al.. **Rede Ciranda: Retratos da infância e juventude: Práticas sociais e abordagens teóricas no município de Assis, SP**. Marília: Fundepe, 2011. 277 p.

XAVIER, Ana Lúcia Pintar et al.. **Rede Ciranda: Desenhando novos caminhos para o trabalho social com crianças e adolescentes**. São Carlos: Pedro & João Editores, 2012. 204 p.