

HELLEN CHRISTINE SERÓDIO THOMAZINHO

**SISTEMA WEB PARA GERENCIAMENTO DO ATENDIMENTO
DE MEDIDAS SÓCIO EDUCATIVAS**

Assis
2010

HELLEN CHRISTINE SERÓDIO THOMAZINHO

**SISTEMA WEB PARA GERENCIAMENTO DO ATENDIMENTO
DE MEDIDAS SÓCIO EDUCATIVAS**

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

Orientador: Dr. Almir Rogério Camolesi.

Área de Concentração: Desenvolvimento de Sistemas

Assis
2010

FICHA CATALOGRÁFICA

THOMAZINHO, Hellen Christine Seródio

Sistema Web para Gerenciamento do Atendimento de Medidas Sócio Educativas /
Hellen Christine Seródio Thomazinho. Fundação Educacional do Município de Assis,
2010.

69 p.

Orientador: Dr. Almir Rogério Camolesi

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

CDD: 001.61
Biblioteca da Fema

SISTEMA WEB PARA GERENCIAMENTO DO ATENDIMENTO DE MEDIDAS SÓCIO EDUCATIVAS

HELLEN CHRISTINE SERÓDIO THOMAZINHO

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: Dr. Almir Rogério Camolesi

Analisadora: Dr^a. Marisa Atsuko Nitto

Assis
2010

DEDICATÓRIA

*Dedico este trabalho a Deus,
a minha mãe Júlia, meu noivo Marcus
e minha amiga Daniela.*

AGRADECIMENTOS

Primeiramente a Deus, é a ele que dirijo minha maior gratidão, mais do que me criar, deu propósito à minha vida. Vem dele tudo o que sou, o que tenho e o que espero.

Ao meu orientador e amigo Dr. Almir Rogério Camolesi, pelas suas orientações, idéias e opiniões, pelo constante estímulo transmitido durante o trabalho.

A minha mãe Julia, por ter me ajudado a concluir mais essa etapa em minha vida, e por representar meu maior exemplo e experiência de amor, amizade e lealdade e, sobretudo, pelo apoio e dedicação.

Ao meu noivo Marcus Felipe pela compreensão, amor e apoio em todos os momentos dessa jornada.

A minha amiga Daniela Antonio Camolesi, pela amizade adquirida nesses 3 anos, pelos momentos difíceis que enfrentamos juntas, e também pelos momentos de alegria. “Amigas de verdade não se separam apenas seguem caminhos diferentes”.

A equipe Integr@Assis pela boa convivência e amizade ao longo desses anos.

E finalmente, agradeço a todos que me ajudaram diretamente ou indiretamente para o desenvolvimento desse trabalho. Muito Obrigado.

RESUMO

Nesse trabalho será apresentada a análise de um sistema de gerenciamento para atendimento de medidas sócio educativas. O sistema proposto tem por objetivo atender e suprir todas as necessidades de uma entidade social, facilitar os atendimentos realizados pela mesma, incluir adolescentes em cursos, e manter um controle de atendimento de medidas sócio educativas em Assis.

O Trabalho apresenta também um estudo sobre as tecnologias que foram utilizadas para o desenvolvimento do sistema, como a linguagem de programação Java, o banco de dados MySQL e a ferramenta de desenvolvimento NetBeans.

Palavras-chave: Java Web, MySQL, NetBeans.

ABSTRACT

In this work will be presents the analysis of a management system to meet socio-educational measures. The proposed system aims to satisfy and meet all the needs of a social entity, to facilitate the care provided by it, including teenagers in courses, and keep track of attendance of socio education in Assis. The work also presents a study on the technologies that were used to develop the system, such as the Java programming language, the MySQL database and development tool NetBeans.

Keywords: Java Web, MySQL, NetBeans.

LISTA DE ILUSTRAÇÕES

Figura 1 - Arquitetura JSF baseada no modelo MVC	20
Figura 2 - Mapa Mental.....	28
Figura 3 - Caso de Uso Geral	30
Figura 4 - UC1 Diagrama de Use Case Cadastrar Usuários.....	31
Figura 5 - UC2 Diagrama de Use Case Cadastrar Escolas	32
Figura 6 - UC3 Diagrama de Use Case Cadastrar Cursos	33
Figura 7 - UC4 Diagrama de Use Case Cadastrar Turmas.....	34
Figura 8 - UC5 Diagrama de Use Case Emitir Relatório por Turma.....	35
Figura 9 - UC6 Diagrama de Use Case Emitir Relatório por Cursos.....	36
Figura 10 - UC7 Diagrama de Use Case Emitir Relatório por Adolescentes	37
Figura 11 - UC8 Diagrama de Use Case Emitir Relatório Adolescentes por Cursos	38
Figura 12 - UC9 Diagrama de Use Case Emitir Relatório Adolescentes por Turma	39
Figura 13 - UC10 Diagrama de Use Case Cadastrar Criança ou Adolescente	40
Figura 14 - UC11 Diagrama de Use Case Inserir Adolescentes em Turmas	41
Figura 15 - UC12 Diagrama de Use Case Inserir total de Atendimentos	42
Figura 16 - UC13 Diagrama de Use Case Inserir Quantidade de Presenças	43
Figura 17 - UC14 Diagrama de Use Case Inserir quantidade Comunidade.....	44
Figura 18 - UC15 Diagrama de Use Case Inserir Frequência Atendidos.....	45
Figura 19 - UC16 Diagrama de Use Case Inserir Frequência Egressos.....	46
Figura 20 - Diagrama de Classes	47
Figura 21 - Diagrama de Sequência Cadastrar Usuário	48
Figura 22 - Diagrama de Sequência Cadastrar Cursos	49
Figura 23 - Diagrama de Sequência Cadastrar Escolas Profissionalizantes	49
Figura 24 - Diagrama de Sequência Cadastrar Turmas	50
Figura 25 - Diagrama de Sequência Cadastrar Adolescentes	50
Figura 26 - Diagrama de Sequência Inserir Adolescentes em Turmas	51
Figura 27 - Diagrama de Sequência Inserir Quantidade de Presença	51
Figura 28 - Diagrama de Sequência Inserir Frequência Comunidade	52
Figura 29 - Diagrama de Sequência Inserir Frequência Atendidos.....	52
Figura 30 - Diagrama de Sequência Inserir Frequência Egressos.....	53
Figura 31 - Diagrama de Atividades	54
Figura 32 - Diagrama Entidade Relacionamento	55
Figura 33 - Work Breakdown Structure.....	56
Figura 34 - Sequenciamento de Atividades	57
Figura 35 – Organização dos Pacotes.....	59
Figura 36 – Organização dos Pacotes.....	60
Figura 37 – Interface do Sistema.....	61
Figura 38 - Menu Cadastros	62
Figura 39 – Tela Cadastrar Adolescentes.....	62
Figura 40 - Tela Consultar Adolescentes.....	63
Figura 41 - Tela Editar Adolescente	64
Figura 42 - Cronograma	69

LISTA DE TABELAS

Tabela 1 - Manter Usuário	31
Tabela 2 - Manter Escolas Profissionalizantes	32
Tabela 3 - Manter Cursos	33
Tabela 4 - Manter Turmas	34
Tabela 5 - Emitir Relatório por Turma.....	35
Tabela 6 - Emitir Relatório por Cursos.....	36
Tabela 7 - Emitir Relatório por Adolescentes.....	37
Tabela 8 - Emitir Relatório Adolescentes por Cursos	38
Tabela 9 - Emitir Relatório Adolescentes por Turmas.....	39
Tabela 10 - Manter Criança ou Adolescente.....	40
Tabela 11 - Inserir Adolescentes em Turma	41
Tabela 12 - Inserir Total de Atendimentos	42
Tabela 13 - Inserir Quantidade de Presenças	43
Tabela 14 - Inserir Frequência Comunidade.....	44
Tabela 15 - Inserir Frequência atendidos	45
Tabela 16 - Inserir Frequência de Egressos	46

LISTA DE ABREVIATURAS E SIGLAS

UML Unified Modeling Language

JSF JavaServer Faces

JSP JavaServer Pages

CSS Cascading Style Sheets

HTML Hyper Text Markup Language (Linguagem de Marcação de Hipertexto)

HTTP HyperText Transport Protocol

MVC Model View Controller

XML Extensible Markup Language (Linguagem de Marcação Extensível)

WBS (Work Breakdown Structure /Estrutura Analítica de Trabalho)

SQL Structure Query Language

JCP Java Community Process

LA Liberdade Assistida

PSC Prestação de Serviço a Comunidade

SUMÁRIO

1 – INTRODUÇÃO	14
1.1– OBJETIVOS.....	15
1.2 – JUSTIFICATIVAS.....	15
1.3 – ESTRUTURA DO TRABALHO.....	15
2-TECNOLOGIAS DE ANÁLISE E DESENVOLVIMENTO	17
2.1- TECNOLOGIAS UTILIZADAS	17
2.1.1– JAVA.....	17
2.1.2 – JAVASERVER PAGES (JSP)	18
2.1.3 – JAVASERVER FACES (JSF).....	19
2.1.4– MYSQL	22
2.1.5 – HIBERNATE.....	22
2.1.6 - NETBEANS.....	23
2.1.7 – TOMCAT	23
2.1.8 – CASCADING STYLE SHEET - CSS	24
2.1.9 – IREPORT	24
2.1.10 – MODEL, VIEW CONTROLLER (MVC)	25
3 - ANÁLISE E ESPECIFICAÇÃO DO SISTEMA	27
3.1 - METODOLOGIA DE ANÁLISE	27
3.2 - LEVANTAMENTO DE REQUISITOS.....	27
3.3 - MAPA MENTAL	28
3.4 - LISTA DE EVENTOS.....	29
3.5 - CASO DE USO.....	30
3.6 - DIAGRAMA DE CLASSES	47
3.7 - DIAGRAMA DE SEQUÊNCIA.....	48
3.8 - DIAGRAMA DE ATIVIDADES.....	54
3.9 - DIAGRAMA ENTIDADE RELACIONAMENTO	55
4 - PROJETO	56
4.1 - WORK BREAKDOWN STRUCTURE	56
4.2 - SEQUENCIAMENTO DE ATIVIDADES.....	57
4.3 – ORÇAMENTO.....	57
4.4 – RECURSOS NECESSÁRIOS PARA O DESENVOLVIMENTO.....	57
4.5 - EQUIPAMENTO	58

5 – IMPLEMENTAÇÃO DA APLICAÇÃO	59
5.1 – ORGANIZAÇÃO DOS PACOTES E CLASSES DO SISTEMA.....	59
5.2 – ORGANIZAÇÃO DAS PÁGINAS DO SISTEMA	60
5.3 – INTERFACE DO SISTEMA.....	61
CONCLUSÃO	65
REFERÊNCIAS.....	67
ANEXO – CRONOGRAMA.....	69

1 – INTRODUÇÃO

A Associação Filantrópica Nosso Lar foi fundada em 25 de dezembro de 1949, em uma proposta para conveniar com a Assistência Social deu o tom para ministrar cursos profissionalizantes. Nos 49 anos seguintes após sua fundação organizou turmas de datilografia, corte e costura e computação, formando o total de 12.995 pessoas da comunidade local até 1998. Atuando no social chegou o reconhecimento comunitário, e novos projetos foram criados.

A partir de Abril de 2003, começou a executar medidas sócio educativas previstas no Estatuto da Criança e Adolescente, de Liberdade Assistida e Prestação de Serviços a Comunidade. Neste seguimento tem atendido 178 jovens encaminhados pelo Poder Judiciário e 50 egressos, serviço este municipalizado. O projeto oferece atendimento técnico para o cumprimento de tais medidas sócio educativas tanto individualmente quanto para grupos e ações de complementaridade organizadas por oficinas.

Desde o seu início o gerenciamento desses atendimentos são feitos manualmente, permitindo que erros humanos ocorram ou até mesmo a perda de informações relevantes, haja vista que, um software traria segurança, agilidade e confiabilidade para a Associação.

Um dos problemas que a Associação tem enfrentado é a falta de um software para o gerenciamento desses dados, surgindo assim à proposta de desenvolver um software para o atendimento de medidas sócio educativas.

O sistema que foi implementado visa suprir todas as necessidades da Associação, manter um maior controle sobre todos os dados referentes a situações de cada criança ou adolescente, armazenar um histórico de atendimentos, incluir esse adolescente em oficinas interna da própria instituição, controlar o termino da medida de cada um, emitir relatórios mensais, entre outros. O mesmo foi desenvolvido direcionado para a internet, pois além de ser de fácil manuseio e atualização, permite ao responsável pelo gerenciamento, ter liberdade de acessar o sistema de qualquer computador e lugar, possibilitando assim uma maior facilidade no controle da Associação, mesmo não estando presente fisicamente na mesma.

1.1 – OBJETIVOS

O Objetivo deste trabalho foi prover um software que facilite os atendimentos realizados pela Associação, integrando informações a respeito do atendimento e do desenvolvimento de crianças e adolescentes de Assis.

1.2 – JUSTIFICATIVAS

Pesquisas realizadas na rede mundial de computadores mostraram a inexistência de um software que gerencie esses tipos de dados, e que permita a sua utilização na internet. Desta forma surgiu à oportunidade de desenvolver o mesmo, a fim de informatizar e gerenciar as informações relacionadas ao atendimento de Medidas Sócio Educativas, para a Associação Filantrópica Nosso Lar. Como o sistema permite o acesso via *WEB*, o responsável por controlar todos os dados, pode acessar o mesmo de qualquer local, podendo assim levantar todos e quaisquer dados referentes aos atendimentos da própria Associação.

As necessidades que levaram o surgimento do sistema foram: os atendimentos serem realizados manualmente, a necessidade de agilidade e proteção desses dados, e a emissão de relatórios mensais.

1.3 – ESTRUTURA DO TRABALHO

Para uma melhor organização, o trabalho foi organizado em capítulos.

No capítulo 2 serão apresentadas as tecnologias utilizadas para a análise e desenvolvimento do sistema, com uma breve descrição de cada uma. A seguir no capítulo 3, as etapas de Análise e Especificações do Sistema contemplando o Levantamento de Requisitos, Lista de Eventos, o Caso de Uso e suas especificações e Diagramas (Classe, Sequência e Entidade Relacionamento). O capítulo 4 mostrará a Work BreakDown Structure(WBS), o Sequenciamento das Atividades e Orçamento do Sistema. O 5º capítulo contém a implementação do sistema, com as divisões de pacotes e de páginas,

mostrando a interface do software. Ao final serão apresentados a conclusão do trabalho, as referências bibliográficas e o cronograma em anexo.

2-TECNOLOGIAS DE ANÁLISE E DESENVOLVIMENTO

Neste capítulo contém a descrição das tecnologias que foram utilizadas para o desenvolvimento do Sistema.

2.1- TECNOLOGIAS UTILIZADAS

Para o desenvolvimento do sistema foi utilizada a linguagem de programação Java com *JavaServer Pages*, usando o framework *JavaServer Faces*. O sistema foi desenvolvido utilizando a ferramenta NetBeans e para persistência com o banco de dados *MySQL*, foi utilizada a ferramenta Hibernate.

2.1.1– JAVA

De tempos em tempos o mundo conhece uma grande mudança. No mundo do desenvolvimento, uma delas aconteceu quando Java foi anunciado, em maio de 1995, pela empresa *Sun Microsystems*, no qual lançava naquele momento uma linguagem de programação que trabalhava em sites produzidos na *World Wide Web*.

O que chamava atenção nessa linguagem era o fato de que ela podia ser portátil para outros sistemas operacionais. Além, sua fama cresceu rapidamente porque a Web como conhecemos hoje estava em ascensão, e Java possibilitava fazer diversas coisas, como animações, que até então não eram possíveis em páginas existentes na *World Wide Web*.

Seu tamanho também era um fator importante, onde depois de instalado a máquina virtual em sua máquina, os arquivos *Applet*, como são conhecidos até hoje, eram pequenos assim como a velocidade de transmissão na época, o que facilitava a visualização desses aplicativos.

O Java amadureceu e com as mudanças ocorridas no mundo, muitas implementações foram criadas, com o intuito de abranger essas mudanças.

Hoje você pode estar usando tanto aplicativos *desktop*, páginas para a Internet ou até mesmo aplicativos pequenos em celulares. Todos criados com a Linguagem Java.

A linguagem Java nos dias de hoje é utilizada por grandes bancos, pois fornece extrema segurança. Também é utilizada por grandes empresas que desejam trafegar uma grande quantidade de dados e necessita de estabilidade e portabilidade entre outras empresas.

As variações de formas e sistemas criados em Java são imensos. Você encontra Java em:

- ✚ Servidores Web.
- ✚ Banco de dados relacionais.
- ✚ Computadores de grande porte.
- ✚ Telefones Móveis.
- ✚ Sistemas de Cartão de crédito entre outros.

(GONÇALVES, Edson 2007).

2.1.2 – JAVASERVER PAGES (JSP)

JavaServer Pages – JSP, é uma tecnologia baseada em Java que simplifica o processo de desenvolvimento de sites da Web dinâmicos. Com JSP, os designers da Web e programadores podem rapidamente incorporar elementos dinâmicos em páginas da Web, utilizando Java embutido e algumas tags de marcação simples. Estas tags fornecem ao designer de HTML um meio de acessar dados e a lógica de negócio armazenados em objetos Java sem ter que dominar as complexidades do desenvolvimento de aplicações.

JavaServer Pages são arquivos textos, normalmente com a extensão ".jsp", que substituem as páginas HTML tradicionais. Os arquivos JSP contêm HTML junto com o código embutido que permitem que o designer de páginas acessem dados do código Java rodando no servidor. Quando a página é solicitada por um usuário e processada pelo servidor HTTP, a parte HTML da

página é então transmitida. No entanto, as partes de código das páginas são executadas no momento em que a solicitação é recebida e o conteúdo dinâmico gerado por este código é unido na página, antes de ser enviado para o usuário. Isto propicia uma separação dos aspectos de apresentação HTML da página, da lógica de programação contida no código.

A tecnologia *JavaServer Pages* foi desenvolvida para facilitar o desenvolvimento de *servlets*, sobretudo considerando grandes e complexas aplicações na Web.

A idéia é que as páginas JSP sejam escritas em vez de páginas HTML, e o código Java e as *tags* HTML estejam mescladas para a formação dessas páginas. Quando a página for requisitada, um *servlet* correspondente é utilizado para responder a requisição, porém o programador não precisa escrever esse *servlet*. Em termos práticos, a utilização de páginas JSP é semelhante à utilização de páginas PHP e ASP (COSTA, Daniel 2008).

2.1.3 – JAVASERVER FACES (JSF)

JavaServer Faces é uma tecnologia que incorpora características de um *framework* MVC para WEB e de um modelo de interfaces gráficas baseado em eventos. Por basear-se no padrão de projeto MVC, uma de suas melhores vantagens é a clara separação entre a visualização e regras de negócio (modelo). A idéia do padrão MVC é dividir uma aplicação em três camadas: modelo, visualização e controle.

O modelo é responsável por representar os objetos de negócio, manter o estado da aplicação e fornecer ao controlador o acesso aos dados. A visualização representa a interface com o usuário, sendo responsável por definir a forma como os dados serão apresentados e encaminhar as ações dos usuários para o controlador. Já a camada de controle é responsável por fazer a ligação entre o modelo e a visualização, além de interpretar as ações do

usuário e as traduzir para uma operação sobre o modelo, onde são realizadas mudanças e, então, gerar uma visualização apropriada.

No *JavaServer Faces*, o controle é composto por um *servlet* denominado *FacesServlet*, por arquivos de configuração e por um conjunto de manipuladores de ações e observadores de eventos. O *FacesServlet* é responsável por receber requisições da WEB, redirecioná-las para o modelo e então remeter uma resposta. Os arquivos de configuração são responsáveis por realizar associações e mapeamentos de ações e pela definição de regras de navegação. Os manipuladores de eventos são responsáveis por receber os dados vindos da camada de visualização, acessar o modelo, e então devolver o resultado para o *FacesServlet*.

O modelo representa os objetos de negócio e executa uma lógica de negócio ao receber os dados vindos da camada de visualização. Finalmente, a visualização é composta por *component trees* (hierarquia de componentes UI), tornando possível unir um componente ao outro para formar interfaces mais complexas. A Figura abaixo mostra a arquitetura do *JavaServer Faces* baseada no modelo MVC.

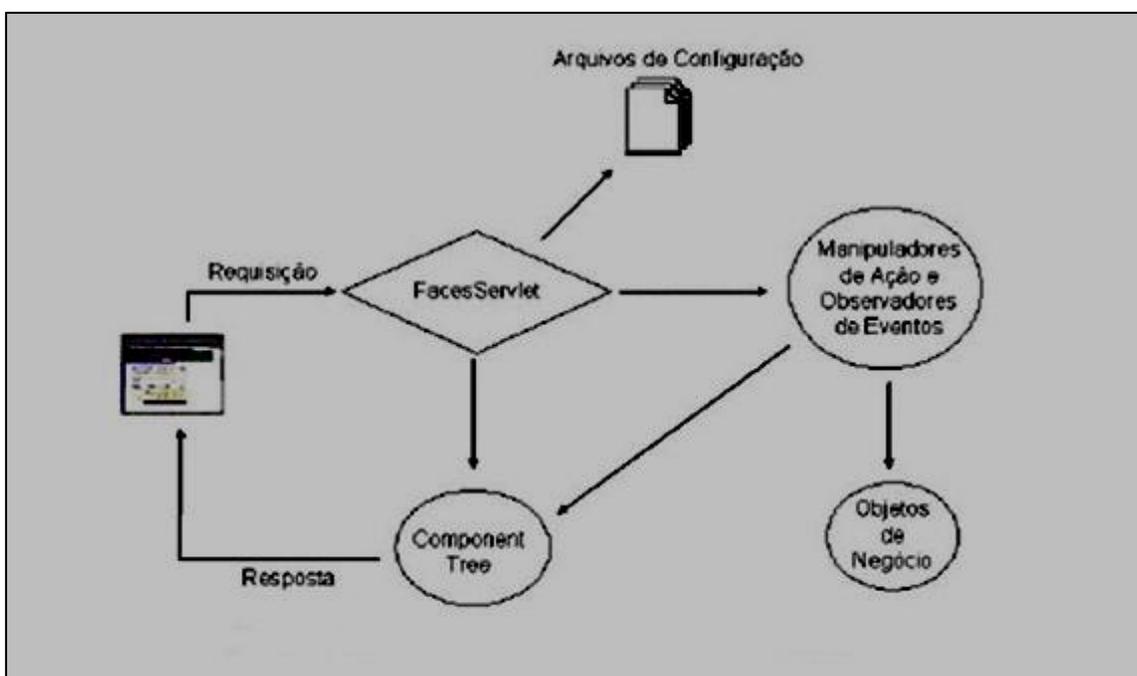


Figura 1 - Arquitetura JSF baseada no modelo MVC

JavaServer Faces oferece ganhos no desenvolvimento de aplicações WEB por diversos motivos:

- ✚ Permite que o desenvolvedor crie UIs através de um conjunto de componentes UIs pré-definidos;
- ✚ Fornece um conjunto de *tags* JSP para acessar os componentes;
- ✚ Reusa componentes da página;
- ✚ Associa os eventos do lado cliente com os manipuladores dos eventos do lado servidor (os componentes de entrada possuem um valor local representando o estado no lado servidor);
- ✚ Fornece separação de funções que envolvem a construção de aplicações WEB.

Embora *JavaServer Faces* forneça *tags* JSP para representar os componentes em uma página, ele foi projetado para ser flexível, sem limitar-se a nenhuma linguagem markup em particular, nem a protocolos ou tipo de clientes. Ele também permite a criação de componentes próprios a partir de classes de componentes, conforme mencionado anteriormente.

JSF possui dois principais componentes: Java APIs para a representação de componentes UI e o gerenciamento de seus estados, manipulação/observação de eventos, validação de entrada, conversão de dados, internacionalização e acessibilidade; e *taglibs* JSP que expressam a interface JSF em uma página JSP e que realizam a conexão dos objetos no lado servidor (TALITA, 2009).

2.1.4– MYSQL

MySQL é um sistema de Gerenciamento de Banco de Dados relacional, multiencadeado, de código – fonte aberto e nível corporativo.

O *MySQL* foi desenvolvido por uma empresa de consultoria na Suécia chamada inicialmente de TcX, depois, com a popularidade do *MySQL*, passou a se chamar *MySQL AB*.

Seu desenvolvimento ocorreu quando estavam precisando de um sistema de banco de dados que fosse extremamente rápido e flexível. Foi, assim então, que eles criaram o *MySQL*, que é vagamente baseado em outro sistema de gerenciamento de banco de dados chamado de *MSQL*.

O *MySQL* é rápido, flexível e confiável. É utilizado em muitos lugares por todo o mundo (GONÇALVES, Edson 2007).

2.1.5 – HIBERNATE

O *Hibernate* é um projeto audacioso que procura ter uma completa solução para o problema de gerenciamento de dados persistentes em Java. O *Hibernate* é um *framework* que se relaciona com o banco de dados, onde esse relacionamento é conhecido como mapeamento objeto/relacional para Java, deixando o desenvolvedor livre para se concentrar em problemas da lógica de negócio.

Hibernate se integra suavemente ao seu sistema se comunicando com o banco de dados como se fosse diretamente feito por sua aplicação. Uma mudança de banco de dados, nesse caso, não se torna traumática, alterando apenas por um detalhe nas configurações do *Hibernate* (GONÇALVES, Edson 2007).

2.1.6 - NETBEANS

NetBeans é um projeto *Open Source* que foi desenvolvido pela *Sun Microsystems* e é um ambiente de desenvolvimento, no qual pode escrever, compilar, depurar e implantar Programas (GONÇALVES, Edson 2007).

2.1.7 – TOMCAT

Servidor Web Java, mais especificamente um *container* de *servlets*. Implementação das tecnologias de *Java Servlet* e *JavaServer Pages*. Desenvolvido em ambiente aberto e participativo, tem a colaboração de todos os desenvolvedores do mundo, mas não pode ser considerado um servidor de aplicação completo, por não preencher todos os requisitos necessários.

O *Tomcat* é robusto e eficiente o suficiente para ser utilizado mesmo num ambiente de produção. Ele tem a capacidade de atuar também como servidor web, ou pode funcionar integrado a um servidor web dedicado como o Apache ou o IIS. Como servidor web, ele provê um servidor web HTTP puramente em Java. O Servidor inclui ferramentas para configuração e gerenciamento, o que também pode ser feito editando-se manualmente arquivos de configuração formatados em XML.

A disponibilização de aplicações nesse servidor deve seguir um padrão rígido e bem estabelecido, garantido que uma mesma aplicação possa ser também montada em outro servidor de aplicação de forma transparente (TOMCAT, 2009).

2.1.8 – CASCADING STYLE SHEET - CSS

A *Cascading Style Sheet (CSS)* é uma ferramenta utilizada para construção da aparência de páginas Web. Permite o uso de uma técnica diferente da convencional (HTML puro), possibilitando uma considerável redução no tempo do trabalho.

O recurso, que traduzido significa Folha de Estilo em Cascata, se tornou uma necessidade para quem deseja ser um bom *webdeveloper* (desenvolvedor de páginas para Internet) e para quem quer criar qualquer projeto para web.

Mas afinal, o que é CSS? Trata-se de uma linguagem para estilos que definem o *layout* de documentos HTML controlando, por exemplo, fontes, cores, margens, linhas, alturas, larguras, imagens de fundo e posicionamentos.

A linguagem HTML também pode ser usada para definir o *layout de websites*. Contudo, a CSS proporciona mais opções, é mais precisa e sofisticada, além de ser suportada pela maioria dos navegadores atuais. A principal diferença entre CSS e HTML é que o primeiro é usado para formatar conteúdos e o segundo para conteúdos já estruturados (SOMERA, 2006).

2.1.9 – IREPORT

IREPORT é um programa *Open Source*, capaz de criar visualmente os mais complexos relatórios para aplicações Java no formato da biblioteca *JasperReports*. Como é escrito em 100% Java, significa que também é multiplataforma. Além disso, seus códigos fontes são distribuídos gratuitamente de acordo com a GNU (General Public License).

Através de uma interface gráfica intuitiva, o desenvolvedor é capaz de criar qualquer tipo de relatório de forma simples e rápida. Mesmo sabendo que o *iReport* desenvolve um formato XML usado pelo *JasperReports*, o que não é difícil de manipular, há uma vantagem em usar esta ferramenta.

Se o desenvolvedor é um usuário iniciante no formato XML do *JasperReports*, o *iReport* supre suas necessidades evitando que seja necessário fazer modificações no código fonte. Caso seja experiente neste formato, o *iReport* minimiza o tempo na criação dos mais complexos relatórios (Gonçalves, Edson, 2008).

2.1.10 – MODEL, VIEW CONTROLLER (MVC)

MVC é um conceito de desenvolvimento e design que tenta separar uma aplicação em três partes distintas. Uma parte, a *Model*, está relacionada ao trabalho atual que a aplicação administra, outra parte, a *View*, está relacionada a exibir dados ou informações dessa aplicação e a terceira parte, *Controller*, em coordenar os dois anteriores exibindo a interface correta ou executando algum trabalho que a aplicação precisa completar.

- ✚ **Model:** O *Model* (Modelo) é o objeto que representa os dados do programa. Maneja esses dados e controlam todas suas transformações. O *Model* são as classes que trabalham no armazenamento e busca de dados. Por exemplo, um cliente pode ser modelado em uma aplicação, e pode haver vários modos de criar novos clientes ou mudar informações de um relativo cliente.
- ✚ **View:** O *View* (Apresentação) é o que maneja a apresentação visual dos dados representados pelo *Model*. Em resumo, é a responsável por apresentar os dados resultantes do *Model* ao usuário. Por exemplo, uma Apresentação poderá ser um local administrativo onde os administradores se logam em uma aplicação. Cada administrador poderá visualizar uma parte do sistema que outro não vê.
- ✚ **Controller:** O *Controller* (Controlador) é o objeto que responde as ordens executadas pelo usuário, atuando sobre os dados apresentados pelo modelo, decidindo como o Modelo deveria ser alterado ou deveria ser revisto e qual apresentação deveria ser exibida. Por exemplo, o Controlador recebe um pedido para exibir uma lista de clientes interagindo com o Modelo e entregando uma Apresentação onde esta a lista poderá ser exibida.

O Modelo MVC é uma forma de desenvolvimento que ajuda na manutenção do sistema, um padrão muito aceito no desenvolvimento de aplicações Java, principalmente no de aplicações escritas para a Web (GONÇALVES, Edson 2008).

3 - ANÁLISE E ESPECIFICAÇÃO DO SISTEMA

Este capítulo contém informações das ferramentas utilizadas para a modelagem do sistema proposto.

3.1 - METODOLOGIA DE ANÁLISE

O sistema foi modelado utilizando a metodologia de Análise Orientada a Objetos. Para essa análise foi utilizado os conceitos de UML (*Unified Modeling Language*) que é uma linguagem de modelagem para documentar e visualizar os artefatos que especificamos e construímos na análise de um sistema (LEE, TEPFENHART 2001).

3.2 - LEVANTAMENTO DE REQUISITOS

Os requisitos foram levantados através de entrevistas com Ana Lucia Pintar Xavier que atualmente é a Coordenadora da Associação Filantrópica Nosso Lar. Durante as entrevistas foram elencados alguns requisitos como:

- ✚ Cadastro de Adolescentes, Família, Cursos.
- ✚ Realização dos Atendimentos com a maior segurança dos Dados.
- ✚ A possibilidade de o sistema listar todos os cursos disponíveis, e o próprio orientador fazer a inclusão dos Adolescentes no mesmo.
- ✚ Relatório dos Adolescentes, Cursos, Turmas, Adolescentes por Turmas, entre outros.

3.3 - MAPA MENTAL

Mapas Mentais, também conhecidos como *Mindmaps*, são considerados um padrão Mundial para criação, gerenciamento e comunicação de idéias. Os mapas apóiam a organização de idéias, de conhecimento por meio de uma visualização intuitiva e amigável, além de possuir grande versatilidade visual. Eles se iniciam com uma idéia central, onde todos os ramos do mapa significam uma decomposição da idéia principal em idéias relacionadas, baseadas em um modelo visual de pensamento (VARGAS, Ricardo 2005).

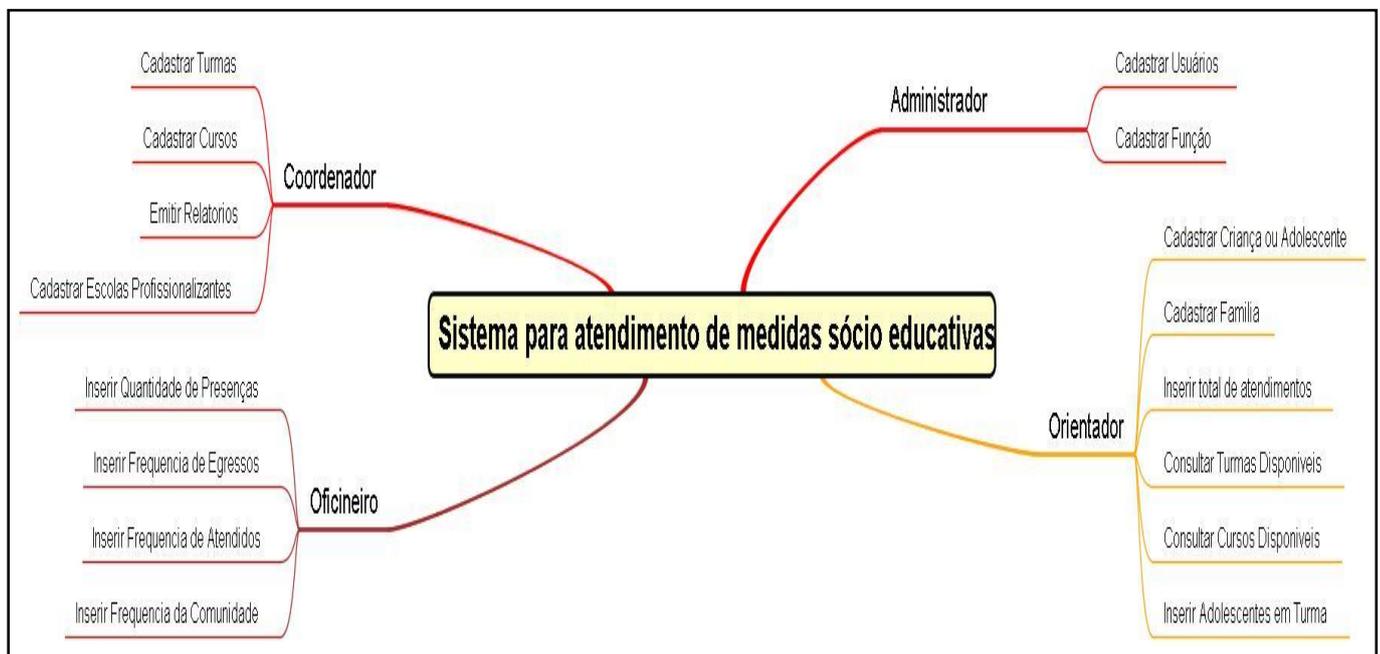


Figura 2 - Mapa Mental

3.4 - LISTA DE EVENTOS

Para modelar o comportamento dos sistemas baseados em objetos, determinam-se quais eventos acontecem. Eventos fazem com que os sistemas tomem várias ações (LEE, TEPFENHART 2001). A seguir são descritos os principais eventos relacionados com as entidades que interagem com o sistema, destacando o que foi desenvolvido:

1. Cadastrar Usuários
2. Cadastrar Função
- 3. Cadastrar Cursos.**
- 4. Cadastrar Turmas**
- 5. Cadastrar Escolas Profissionalizantes.**
6. Emitir Relatórios de Adolescentes.
7. Emitir Relatórios de Cursos.
8. Emitir Relatórios de Turmas.
9. Emitir Relatórios Adolescentes por Turmas.
- 10. Inserir Adolescentes em Turmas.**
- 11. Inserir Total de Atendimentos.**
- 12. Consultar Cursos Disponíveis.**
- 13. Consultar Turmas Disponíveis.**
- 14. Cadastrar Família.**
- 15. Cadastrar Crianças ou Adolescentes.**
- 16. Inserir quantidade de presenças (turmas).**
- 17. Inserir Frequência da Comunidade.**
- 18. Inserir Frequência de atendidos (LA E PSC) em Cursos.**
- 19. Inserir Frequência de Egressos em Cursos.**

3.5 - CASO DE USO

Um caso de uso especifica uma sequência de ações, inclusive variantes, que um sistema realiza e que produz um observável resultado de valor para um particular ator (LEE, TEPFENHART 2001).

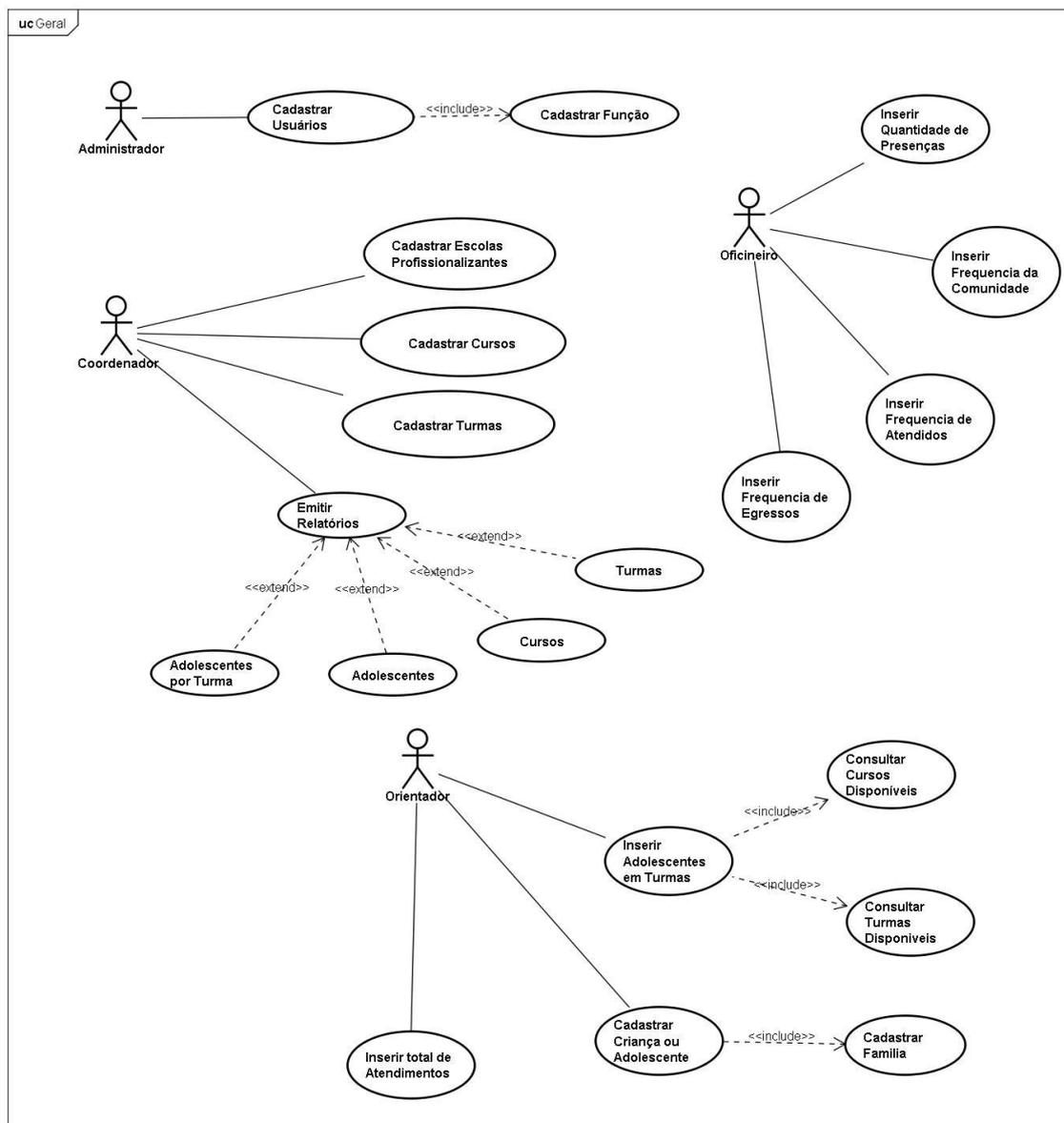


Figura 3 - Caso de Uso Geral

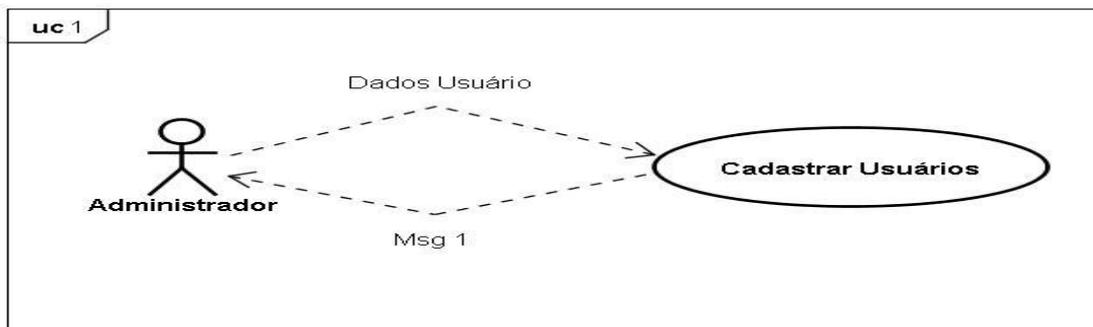


Figura 4 - UC1 Diagrama de Use Case Cadastrar Usuários

Nome da Use Case 1	Cadastrar Usuário
Ator (es)	Administrador
Pré-Condição	O administrador deverá estar autenticado no sistema.
Cenário Principal	<ol style="list-style-type: none"> 1- O sistema solicita os dados necessários para o cadastro do usuário. 2- O administrador informa os dados de acordo com os campos a serem preenchidos. 3- O sistema solicita os dados para o cadastro da função. 4- O administrador informa os dados necessários. 5- O administrador seleciona a opção Cadastrar 6- O sistema emite a mensagem "Usuário cadastrado com sucesso". 7- O sistema cadastra o usuário.
Cenário Alternativo	<p>Se o administrador não informar os dados para o cadastro da função, o sistema informa que o usuário não esta cadastrado.</p> <p>O administrador poderá cancelar o processo durante o cadastro.</p>
Casos de Teste	<ol style="list-style-type: none"> 4.1- O sistema verifica se os campos foram preenchidos corretamente. 4.2- O sistema não confirma o cadastro e emite uma mensagem de erro. 4.3- O sistema cancela a operação.

Tabela 1 - Manter Usuário

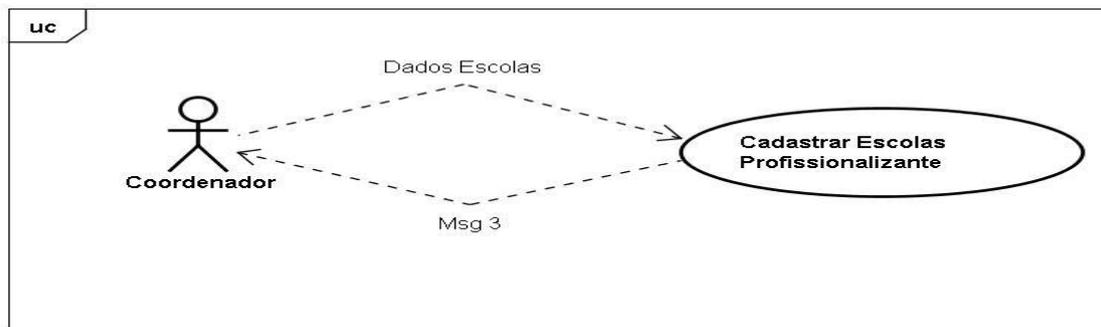


Figura 5 - UC2 Diagrama de Use Case Cadastrar Escolas

Nome da Use Case 2	Cadastrar Escolas Profissionalizantes
Ator (es)	Coordenador
Pré-Condição	O Coordenador deverá estar autenticado no sistema.
Cenário Principal	<ol style="list-style-type: none"> 1- O sistema solicita os dados necessários para o cadastro das Escolas Profissionalizantes. 2- O coordenador informa os dados de acordo com os campos a serem preenchidos. 3- O coordenador seleciona o botão cadastrar. 4- O sistema emite a mensagem “Escolas cadastradas com sucesso”. 5- O sistema cadastrar as Escolas.
Cenário Alternativo	O coordenador poderá cancelar o processo durante o cadastro.
Casos de Teste	<ol style="list-style-type: none"> 3.1- O sistema não confirma o cadastro e emite uma mensagem de erro para o Coordenador. 3.2- O sistema cancela a operação.

Tabela 2 - Manter Escolas Profissionalizantes

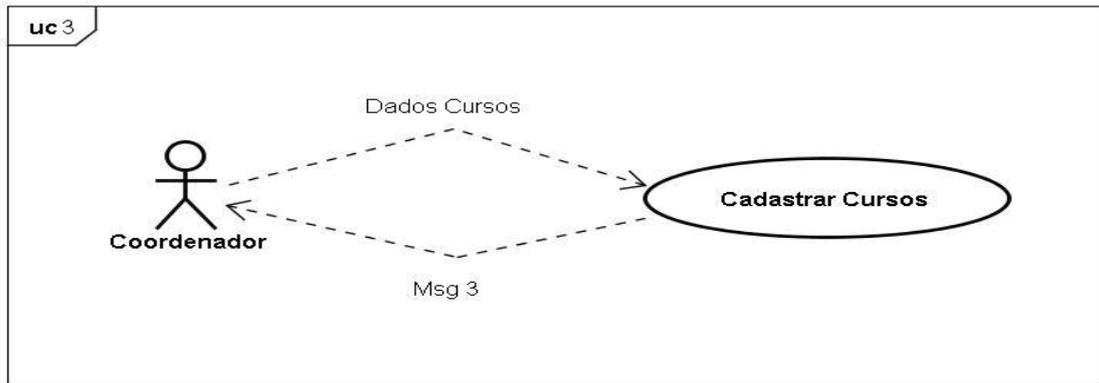


Figura 6 - UC3 Diagrama de Use Case Cadastrar Cursos

Nome da Use Case 3	Cadastrar Cursos
Ator (es)	Coordenador
Pré-Condição	O coordenador deverá estar autenticado no sistema.
Cenário Principal	<p>1- O sistema solicita os dados necessários para o cadastramento dos cursos.</p> <p>2- O coordenador informa os dados de acordo com os campos a serem preenchidos.</p> <p>3- O coordenador seleciona o botão cadastrar.</p> <p>4- O sistema emite a mensagem “Cursos cadastrados com sucesso”.</p> <p>5- O sistema cadastra os cursos.</p>
Cenário Alternativo	O coordenador poderá cancelar o processo durante o cadastro.
Casos de Teste	<p>3.1- O sistema não confirma o cadastro e emite uma mensagem de erro.</p> <p>3.2- O sistema cancela a operação.</p>

Tabela 3 - Manter Cursos

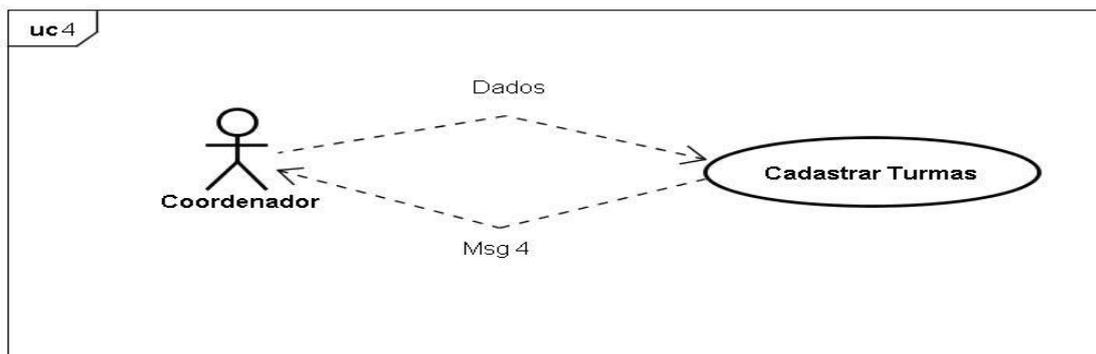


Figura 7 - UC4 Diagrama de Use Case Cadastrar Turmas

Nome da Use Case 4	Cadastrar Turmas
Ator (es)	Coordenador
Pré-Condição	O coordenador deverá estar autenticado no sistema.
Cenário Principal	<p>1- O sistema solicita os dados necessários para o cadastro das Turmas.</p> <p>2- O coordenador informa os dados de acordo com os campos a serem preenchidos.</p> <p>3- O coordenador seleciona o botão cadastrar.</p> <p>4- O sistema emite a mensagem “Turmas cadastradas com sucesso”.</p> <p>5- O sistema cadastra as Turmas.</p>
Cenário Alternativo	O coordenador poderá cancelar o processo durante o cadastro.
Casos de Teste	<p>3.1- O sistema não confirma o cadastro e emite uma mensagem de erro.</p> <p>3.2- O sistema cancela a operação.</p>

Tabela 4 - Manter Turmas

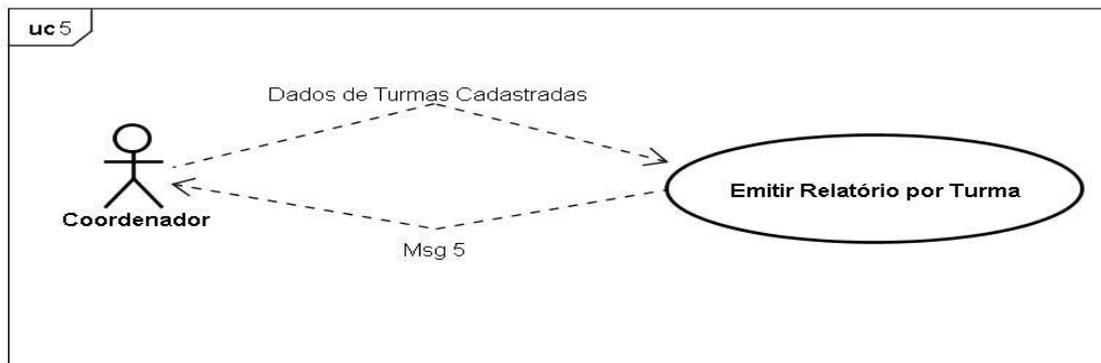


Figura 8 - UC5 Diagrama de Use Case Emitir Relatório por Turma

Nome da Use Case 5	Emitir Relatório por Turma
Ator (es)	Coordenador
Pré-Condição	O coordenador deverá estar autenticado no sistema.
Descrição	Os relatórios serão impressos conforme solicitados.
Cenário Principal	<p>1- O sistema disponibiliza os dados necessários para o relatório.</p> <p>2- O coordenador seleciona o botão Visualizar Relatório.</p> <p>3- O coordenador seleciona o botão Imprimir.</p> <p>4- O sistema imprime o relatório com sucesso.</p>
Cenário Alternativo	O coordenador poderá visualizar o relatório e não imprimir.
Casos de Teste	4.1- O coordenador cancela a operação.

Tabela 5 - Emitir Relatório por Turma

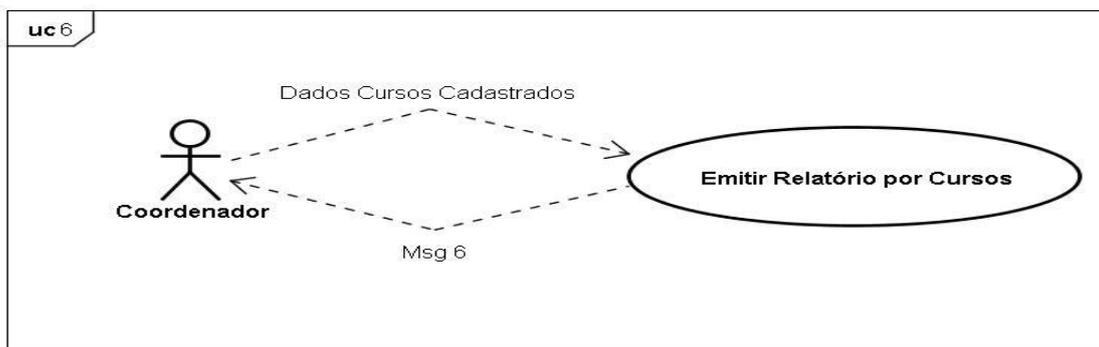


Figura 9 - UC6 Diagrama de Use Case Emitir Relatório por Cursos

Nome da Use Case 6	Emitir Relatório por Cursos
Ator (es)	Coordenador
Pré-Condição	O coordenador deverá estar autenticado no sistema.
Descrição	Os relatórios serão impressos conforme solicitados.
Cenário Principal	<ol style="list-style-type: none"> 1- O sistema disponibiliza os dados necessários para o relatório. 2- O coordenador seleciona o botão Visualizar Relatório. 3- O coordenador seleciona o botão Imprimir. 4- O sistema imprime o relatório com sucesso.
Cenário Alternativo	O coordenador poderá visualizar o relatório e não imprimir.
Casos de Teste	4.1- O coordenador cancela a operação.

Tabela 6 - Emitir Relatório por Cursos

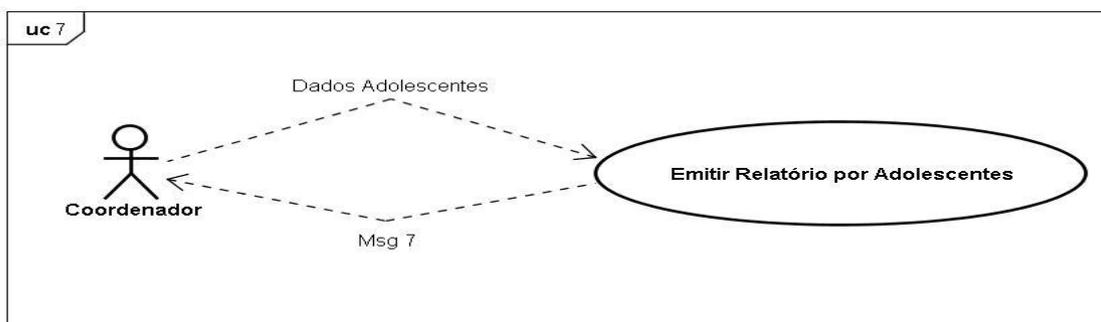


Figura 10 - UC7 Diagrama de Use Case Emitir Relatório por Adolescentes

Nome da Use Case 7	Emitir Relatório por Adolescentes
Ator (es)	Coordenador
Pré-Condição	O coordenador deverá estar autenticado no sistema.
Descrição	Os relatórios serão impressos conforme solicitados.
Cenário Principal	<ol style="list-style-type: none"> 1- O sistema disponibiliza os dados necessários para o relatório. 2- O coordenador seleciona o botão Visualizar Relatório. 3- O coordenador seleciona o botão Imprimir. 4- O sistema imprime o relatório com sucesso.
Cenário Alternativo	O coordenador poderá visualizar o relatório e não imprimir.
Casos de Teste	4.1- O coordenador cancela a operação.

Tabela 7 - Emitir Relatório por Adolescentes

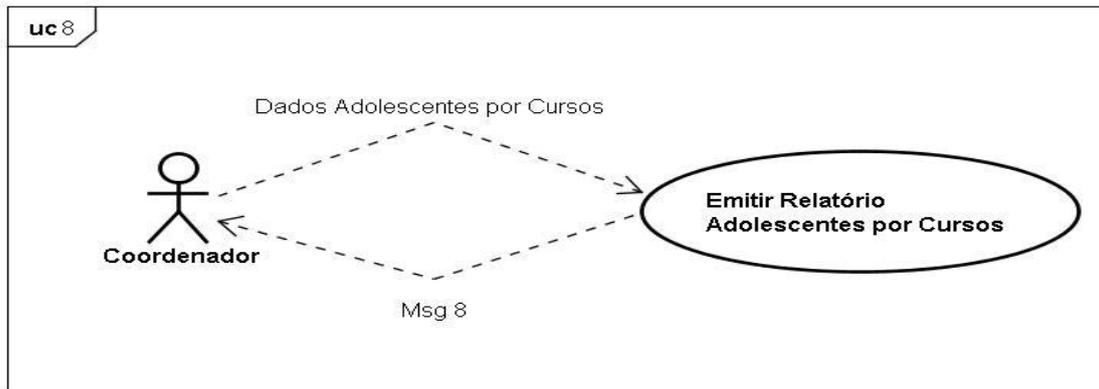


Figura 11 - UC8 Diagrama de Use Case Emitir Relatório Adolescentes por Cursos

Nome da Use Case 8	Emitir Relatório Adolescentes por Cursos
Ator (es)	Coordenador
Pré-Condição	O coordenador deverá estar autenticado no sistema.
Descrição	Os relatórios serão impressos conforme solicitados.
Cenário Principal	<ol style="list-style-type: none"> 1- O sistema disponibiliza os dados necessários para o relatório. 2- O coordenador seleciona o botão Visualizar Relatório. 3- O coordenador seleciona o botão Imprimir. 4- O sistema imprime o relatório com sucesso.
Cenário Alternativo	O coordenador poderá visualizar o relatório e não imprimir.
Casos de Teste	4.1- O coordenador cancela a operação.

Tabela 8 - Emitir Relatório Adolescentes por Cursos

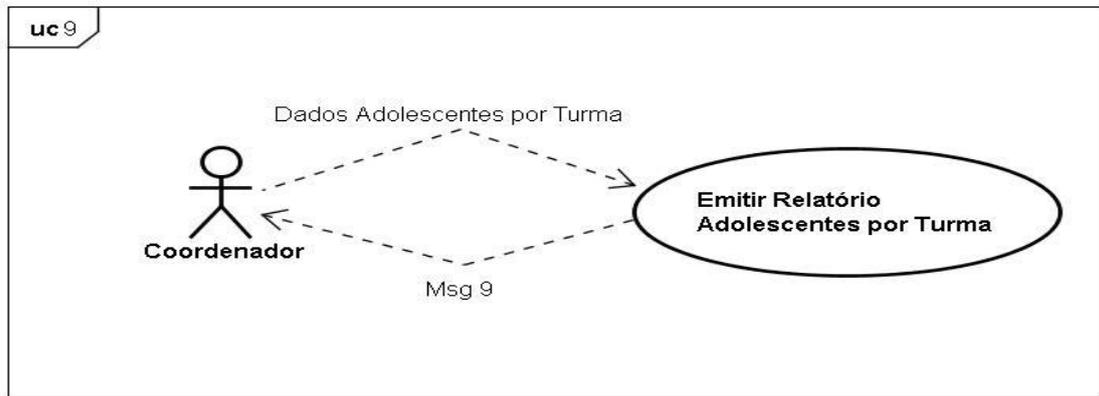


Figura 12 - UC9 Diagrama de Use Case Emitir Relatório Adolescentes por Turma

Nome da Use Case 9	Emitir Relatório Adolescentes por Turmas
Ator (es)	Coordenador
Pré-Condição	O coordenador deverá estar autenticado no sistema.
Descrição	Os relatórios serão impressos conforme solicitados.
Cenário Principal	<ol style="list-style-type: none"> 1- O sistema disponibiliza os dados necessários para o relatório. 2- O coordenador seleciona o botão Visualizar Relatório. 3- O coordenador seleciona o botão Imprimir. 4- O sistema imprime o relatório com sucesso.
Cenário Alternativo	O coordenador poderá visualizar o relatório e não imprimir.
Casos de Teste	4.1- O coordenador cancela a operação.

Tabela 9 - Emitir Relatório Adolescentes por Turmas

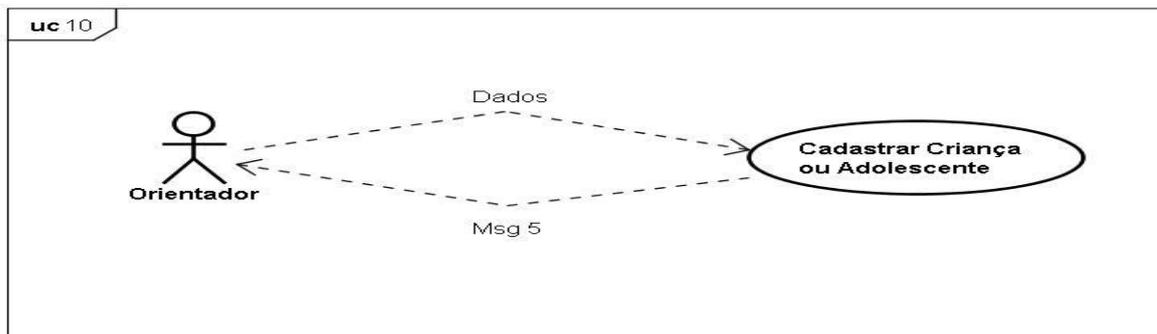


Figura 13 - UC10 Diagrama de Use Case Cadastrar Criança ou Adolescente

Nome da Use Case 10	Cadastrar Criança ou Adolescente
Ator (es)	Orientador
Pré-Condição	O orientador deverá estar autenticado no sistema.
Cenário Principal	<p>1- O sistema solicita os dados necessários para o cadastro.</p> <p>2- O orientador informa os dados de acordo com os campos a serem preenchidos.</p> <p>3- O orientador seleciona o botão cadastrar.</p> <p>4- O sistema solicita os dados para o cadastro da Família.</p> <p>5- O orientador informa os dados de acordo com os campos a serem preenchidos.</p> <p>6- O sistema emite a mensagem “Adolescente cadastrado com sucesso”.</p> <p>7- O sistema efetua o cadastro.</p>
Cenário Alternativo	O orientador poderá cancelar o processo durante o cadastro.
Casos de Teste	<p>4.1- O sistema verifica se os campos foram preenchidos corretamente.</p> <p>4.2- O sistema não confirma o cadastro e emite uma mensagem de erro.</p> <p>4.3- O sistema cancela a operação.</p>

Tabela 10 - Manter Criança ou Adolescente

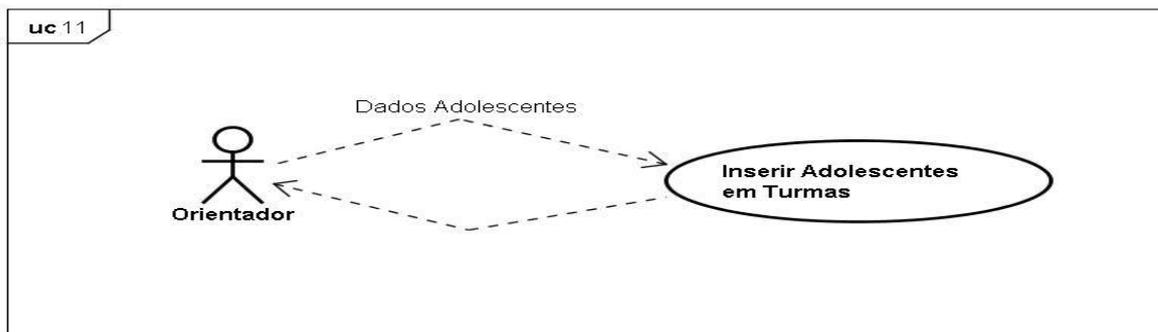


Figura 14 - UC11 Diagrama de Use Case Inserir Adolescentes em Turmas

Nome da Use Case 11	Inserir Adolescentes em Turmas
Ator (es)	Orientador
Pré-Condição	O orientador deverá estar autenticado no sistema.
Cenário Principal	<p>1- O Orientador seleciona a opção Consultar :</p> <ul style="list-style-type: none"> • Cursos Disponíveis. • Turmas Disponíveis. <p>2- O sistema faz a busca e verifica os Cursos Disponíveis e as Turmas Disponíveis.</p> <p>3- O sistema solicita os dados necessários para o cadastro.</p> <p>4- O orientador informa os dados de acordo com os campos a serem preenchidos.</p> <p>5- O orientador seleciona o botão inserir.</p> <p>6- O sistema emite a mensagem “Adolescente inserido com sucesso”.</p> <p>7- O sistema insere o adolescente em uma Turma.</p>
Cenário Alternativo	O orientador poderá cancelar o processo durante o cadastro.
Casos de Teste	<p>4.1- O sistema verifica se os campos foram preenchidos corretamente.</p> <p>4.2- O sistema não confirma o cadastro e emite uma mensagem de erro.</p> <p>4.3- O sistema cancela a operação.</p>

Tabela 11 - Inserir Adolescentes em Turma

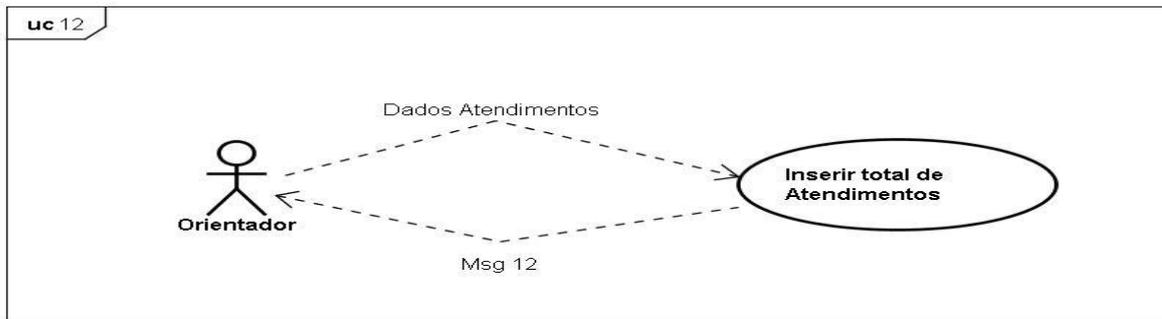


Figura 15 - UC12 Diagrama de Use Case Inserir total de Atendimentos

Nome da Use Case 12	Inserir total de atendimentos (Mensal)
Ator (es)	Orientador
Pré-Condição	O orientador deverá estar autenticado no sistema.
Cenário Principal	<ol style="list-style-type: none"> 1- O orientador deverá estar na página inicial do sistema. 2- O orientador informa os dados de acordo com os campos a serem preenchidos. 3- O orientador seleciona o botão Salvar. 4- O sistema emite a mensagem "Item salvo com sucesso". 5- O sistema salva as informações.
Cenário Alternativo	O orientador poderá cancelar o processo durante o cadastro.
Casos de Teste	<ol style="list-style-type: none"> 3.1- O sistema não confirma o cadastro e emite uma mensagem de erro. 3.2- O sistema cancela a operação.

Tabela 12 - Inserir Total de Atendimentos

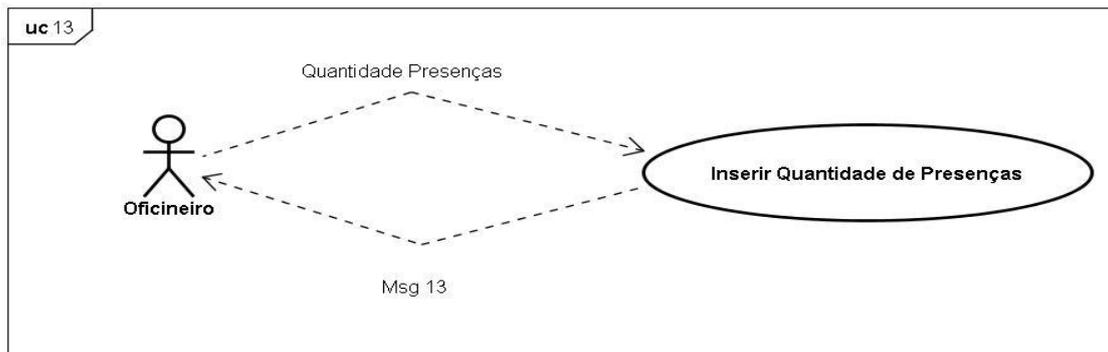


Figura 16 - UC13 Diagrama de Use Case Inserir Quantidade de Presenças

Nome da Use Case 13	Inserir Quantidade de Presenças
Ator (es)	Oficineiro
Pré-Condição	O oficineiro deverá estar autenticado no sistema.
Cenário Principal	<p>1- O sistema solicita os dados necessários para inserir a quantidade de presenças.</p> <p>2- O oficineiro informa os dados de acordo com os campos a serem preenchidos.</p> <p>3- O oficineiro seleciona o botão Salvar.</p> <p>4- O sistema emite a mensagem "Itens salvos com sucesso".</p> <p>5- O sistema armazena as informações.</p>
Cenário Alternativo	O oficineiro poderá cancelar o processo durante o cadastro.
Casos de Teste	<p>3.1- O sistema não confirma as informações e emite uma mensagem de erro.</p> <p>3.2- O sistema cancela a operação.</p>

Tabela 13 - Inserir Quantidade de Presenças

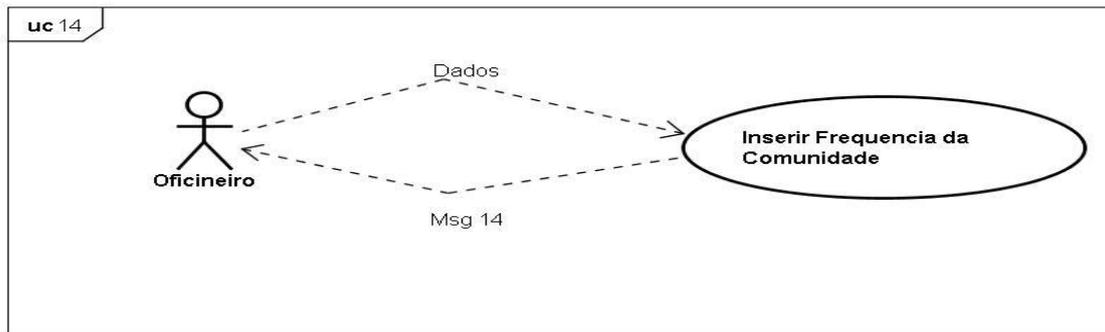


Figura 17 - UC14 Diagrama de Use Case Inserir quantidade Comunidade

Nome da Use Case 14	Inserir Frequência da Comunidade
Ator (es)	Oficineiro
Pré-Condição	O Oficineiro deverá estar autenticado no sistema.
Cenário Principal	<p>1- O sistema solicita os dados necessários para inserir a quantidade.</p> <p>2- O oficineiro informa os dados de acordo com os campos a serem preenchidos.</p> <p>3- O Oficineiro seleciona o botão Salvar.</p> <p>4- O sistema emite a mensagem “Informações salvas com sucesso”.</p> <p>5- O sistema salva as informações.</p>
Cenário Alternativo	O oficineiro poderá cancelar o processo.
Casos de Teste	<p>3.1- O sistema não confirma as informações e emite uma mensagem de erro.</p> <p>3.2- O sistema cancela a operação.</p>

Tabela 14 - Inserir Frequência Comunidade

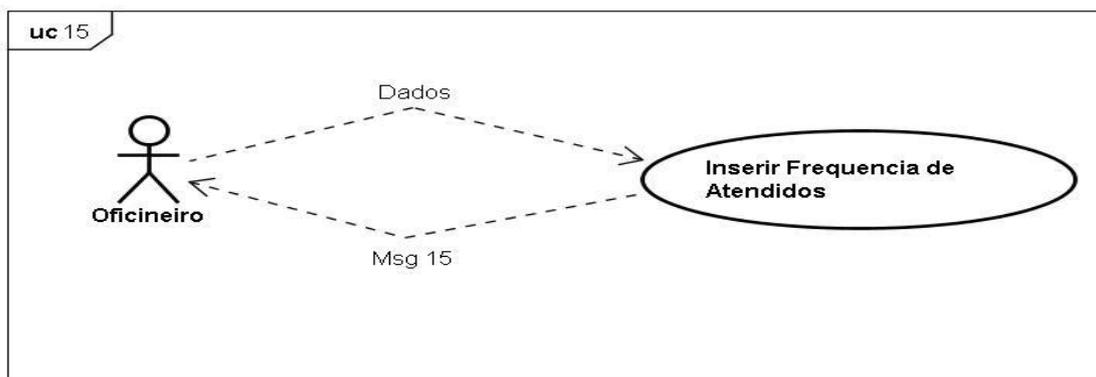


Figura 18 - UC15 Diagrama de Use Case Inserir Frequência Atendidos

Nome da Use Case 15	Inserir Frequência de Atendidos
Ator (es)	Oficineiro
Pré-Condição	O oficineiro deverá estar autenticado no sistema.
Cenário Principal	<p>1- O sistema solicita os dados necessários para inserir as informações.</p> <p>2- O oficineiro informa os dados de acordo com os campos a serem preenchidos.</p> <p>3- O oficineiro seleciona o botão Salvar.</p> <p>4- O sistema emite a mensagem “Informações salvas com sucesso”.</p> <p>5- O sistema salva as informações.</p>
Cenário Alternativo	O oficineiro poderá cancelar o processo durante o cadastro.
Casos de Teste	<p>3.1- O sistema não confirma as informações e emite uma mensagem de erro.</p> <p>3.2- O sistema cancela a operação.</p>

Tabela 15 - Inserir Frequência atendidos

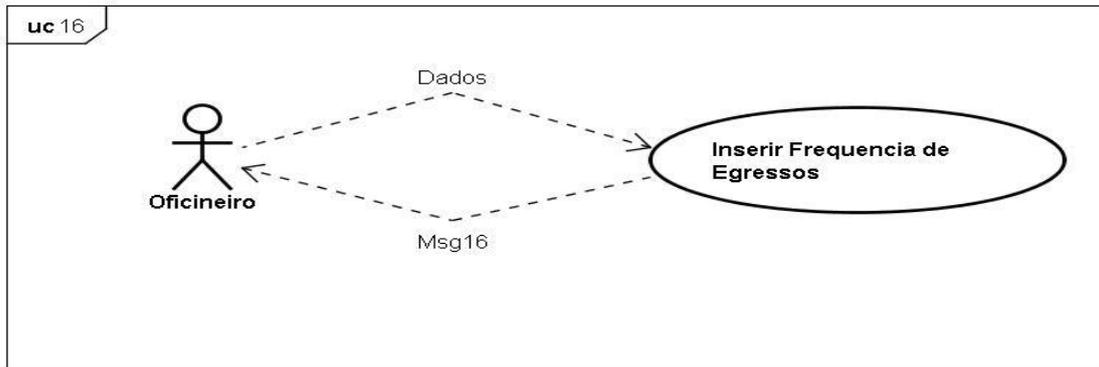


Figura 19 - UC16 Diagrama de Use Case Inserir Frequência Egressos

Nome da Use Case 16	Inserir Frequência de Egressos
Ator (es)	Oficineiro
Pré-Condicion	O oficineiro deverá estar autenticado no sistema.
Cenário Principal	<p>1- O sistema solicita os dados necessários para cadastrar as informações.</p> <p>2- O oficineiro informa os dados de acordo com os campos a serem preenchidos.</p> <p>3- O Oficineiro seleciona o botão Salvar.</p> <p>4- O sistema emite a mensagem “Informações salvas com sucesso”.</p> <p>5- O sistema salvas as informações.</p>
Cenário Alternativo	O oficineiro poderá cancelar o processo.
Casos de Teste	<p>3.1- O sistema não confirma as informações e emite uma mensagem de erro.</p> <p>3.2- O sistema cancela a operação.</p>

Tabela 16 - Inserir Frequência de Egressos

3.6 - DIAGRAMA DE CLASSES

Um Diagrama de Classes descreve os tipos de objetos no sistema e os vários tipos de relacionamentos estático que existem entre eles. O Diagrama de classe mostra também atributos e operações de uma classe e as restrições à maneira com que os objetos são conectados (BOOCH, JACOBSON, RUMBAUGH,2000).

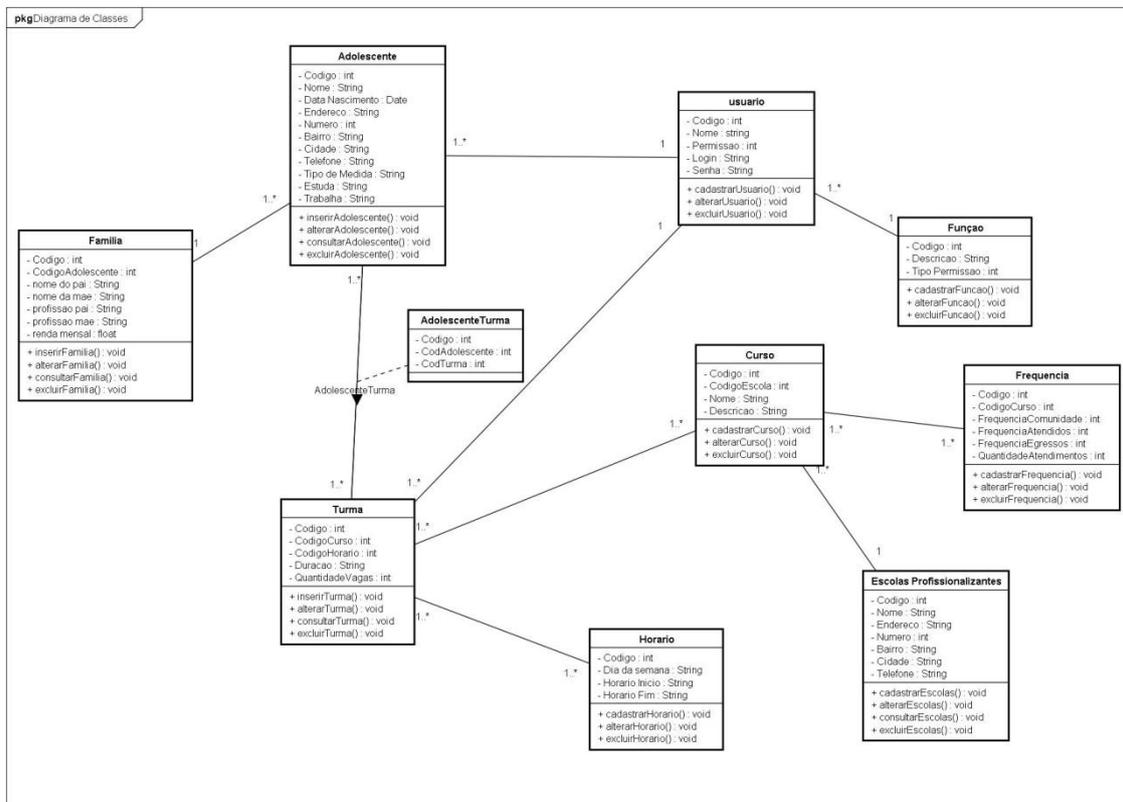


Figura 20 - Diagrama de Classes

3.7 - DIAGRAMA DE SEQUÊNCIA

Um diagrama de sequência mostra o fluxo de controle por ordenamento de tempo e enfatiza a passagem de mensagens à medida que elas vão se desenrolando durante o tempo. Ele revela a sequência de mensagens que implementam um serviço ou transação (LEE, TEPFENHART 2001).

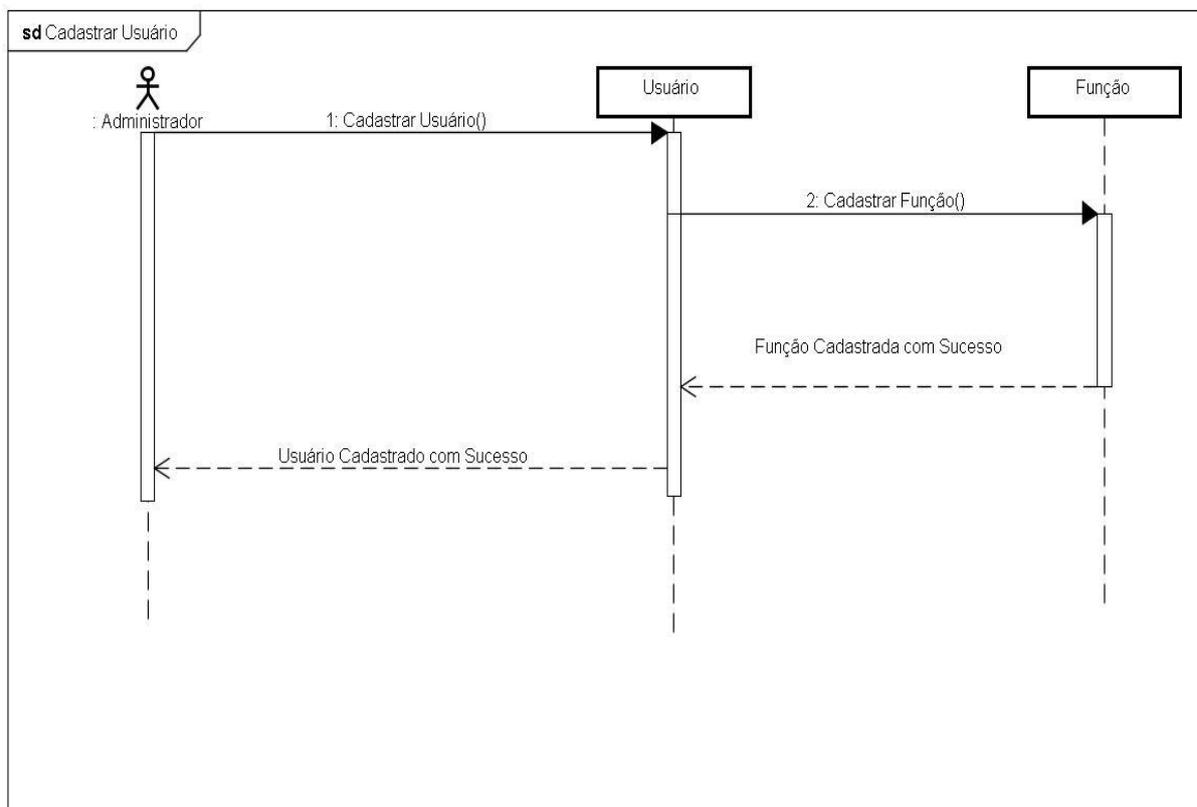


Figura 21 - Diagrama de Sequência Cadastrar Usuário

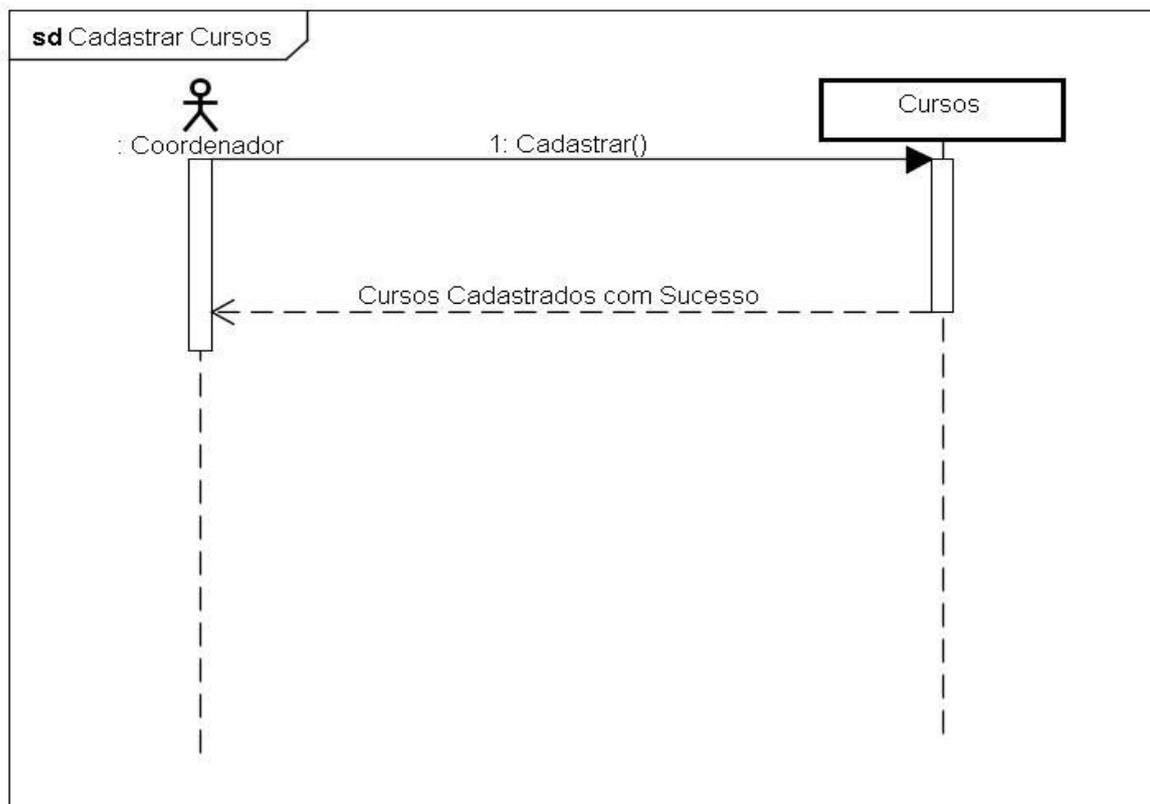


Figura 22 - Diagrama de Sequência Cadastrar Cursos

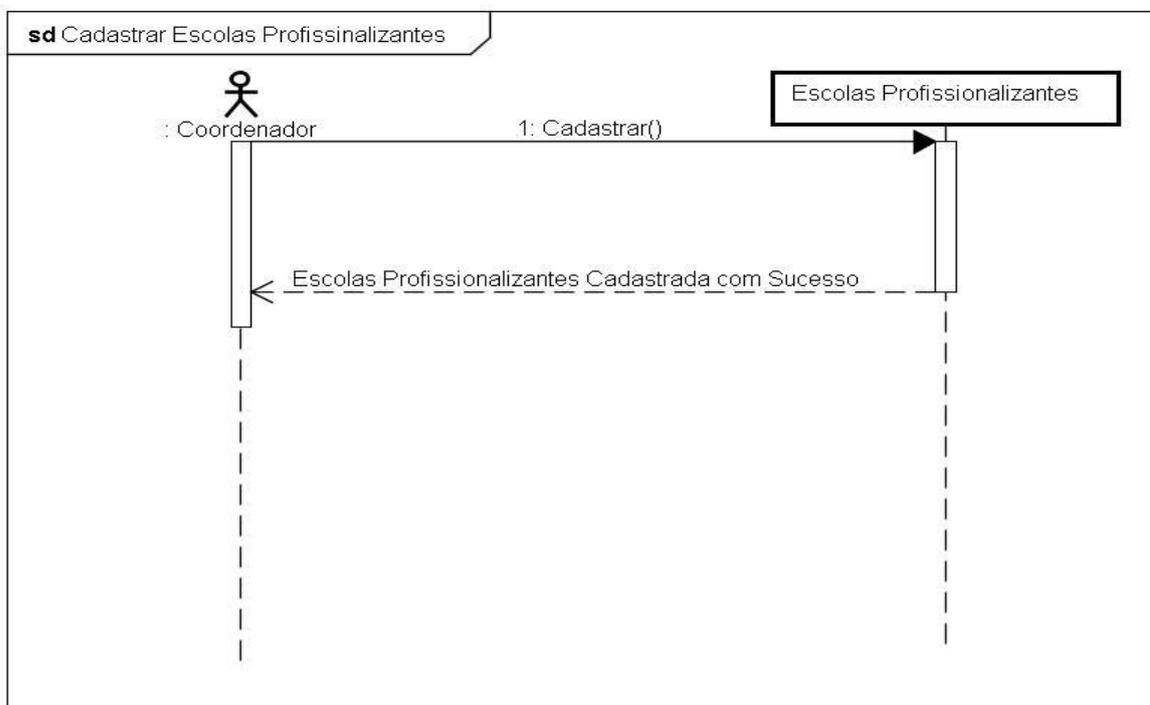


Figura 23 - Diagrama de Sequência Cadastrar Escolas Profissinalizantes

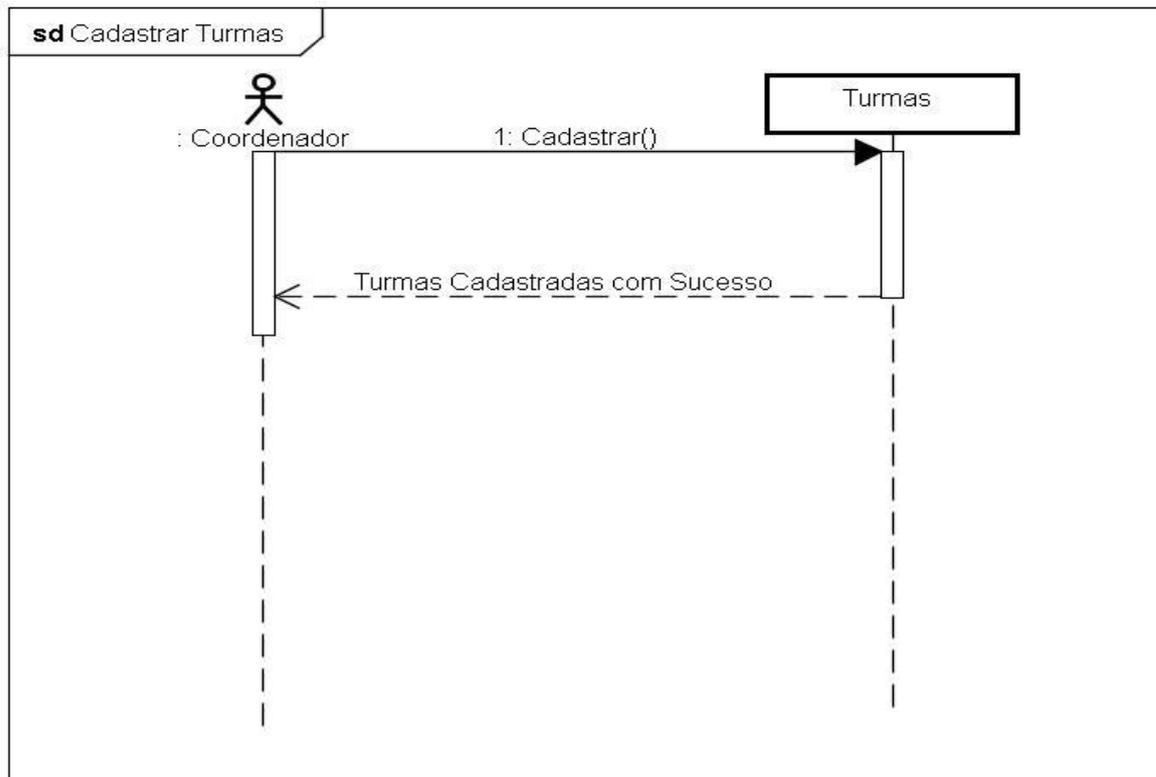


Figura 24 - Diagrama de Sequência Cadastrar Turmas

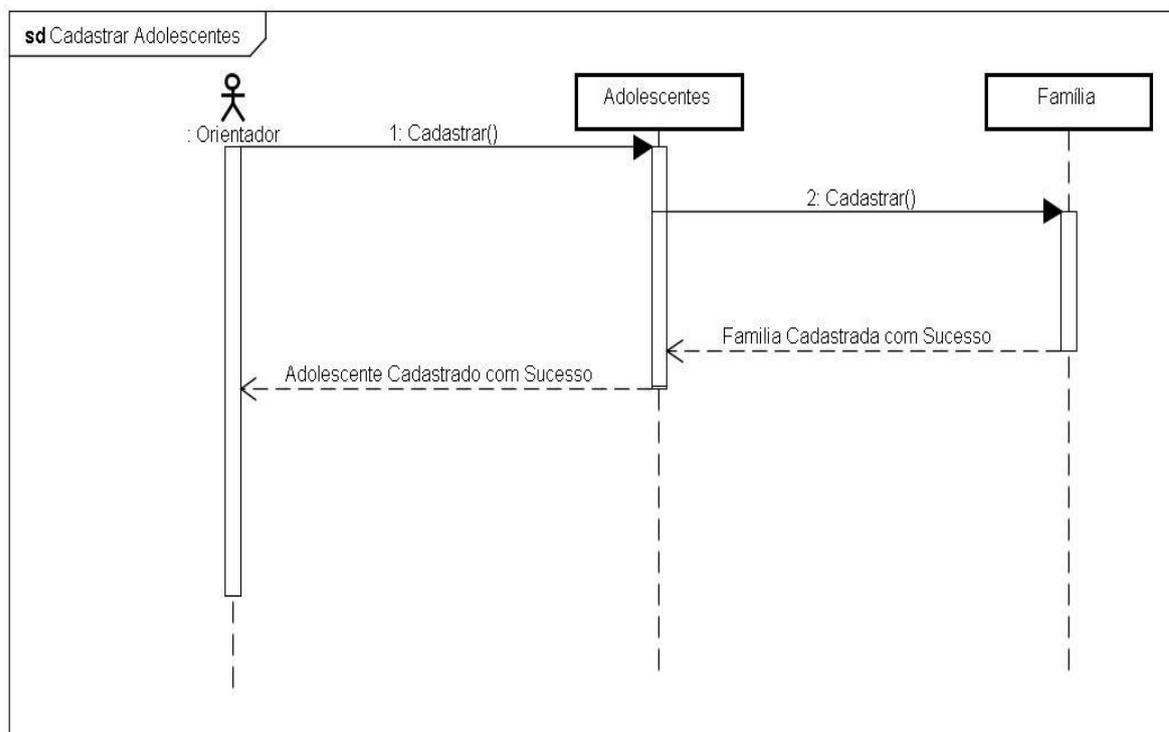


Figura 25 - Diagrama de Sequência Cadastrar Adolescentes

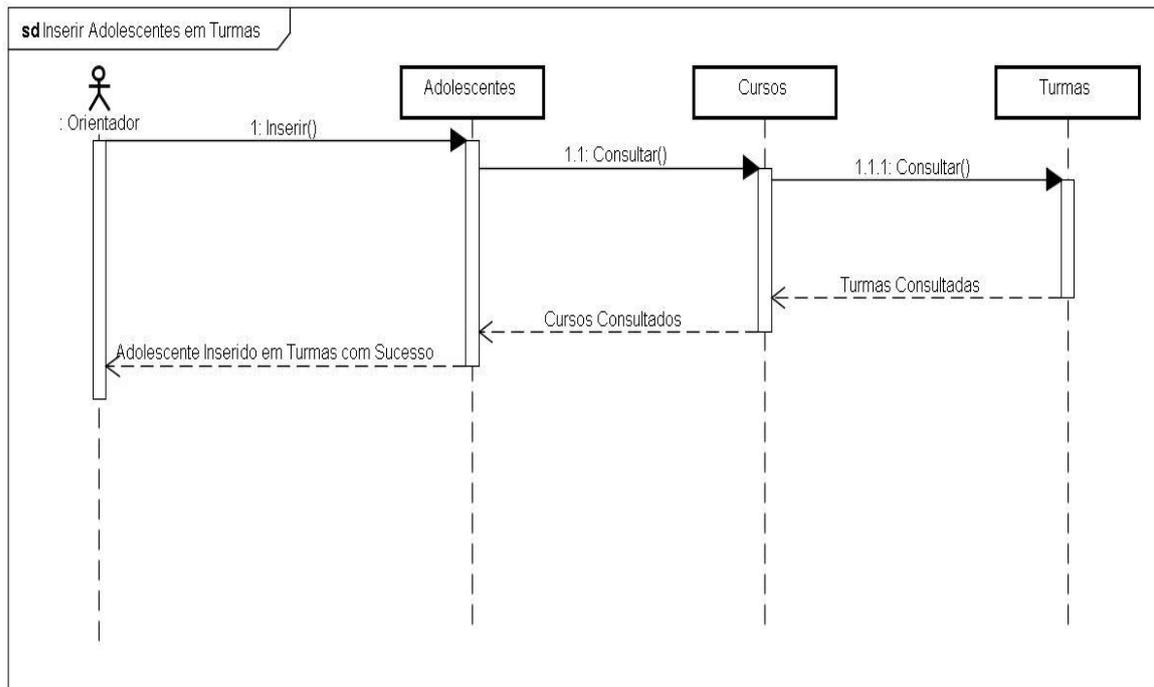


Figura 26 - Diagrama de Sequência Inserir Adolescentes em Turmas

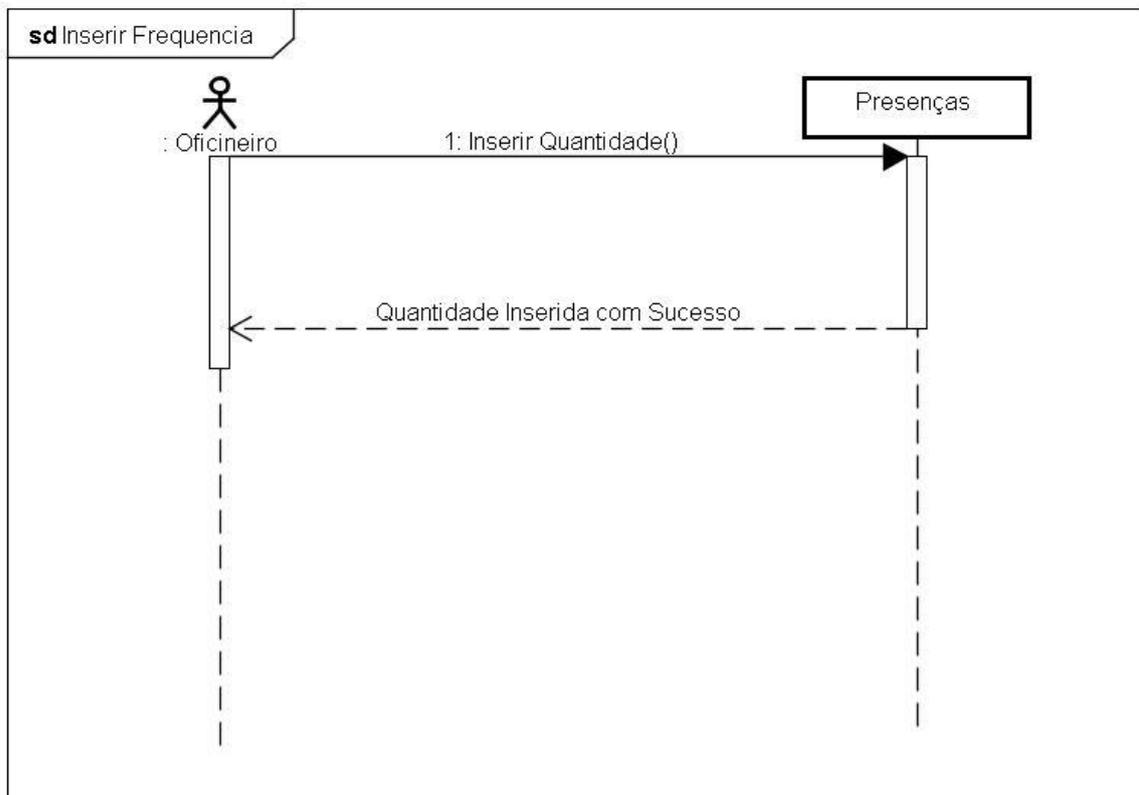


Figura 27 - Diagrama de Sequência Inserir Quantidade de Presença

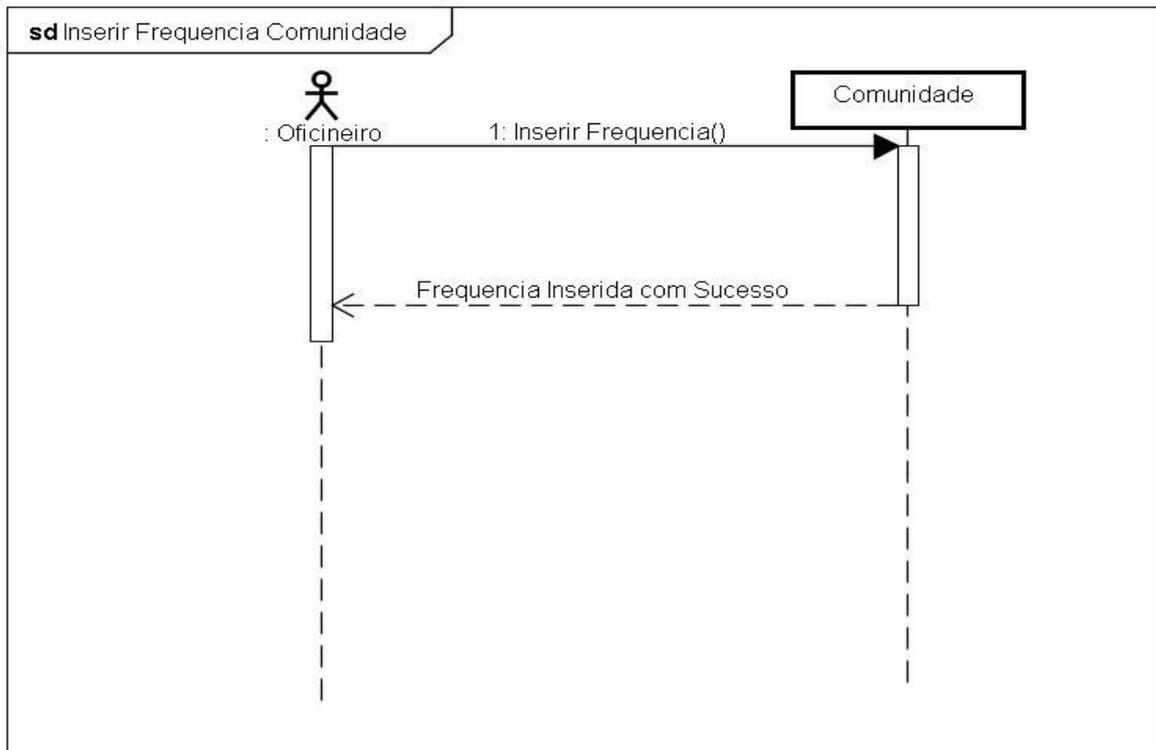


Figura 28 - Diagrama de Sequência Inserir Frequência Comunidade

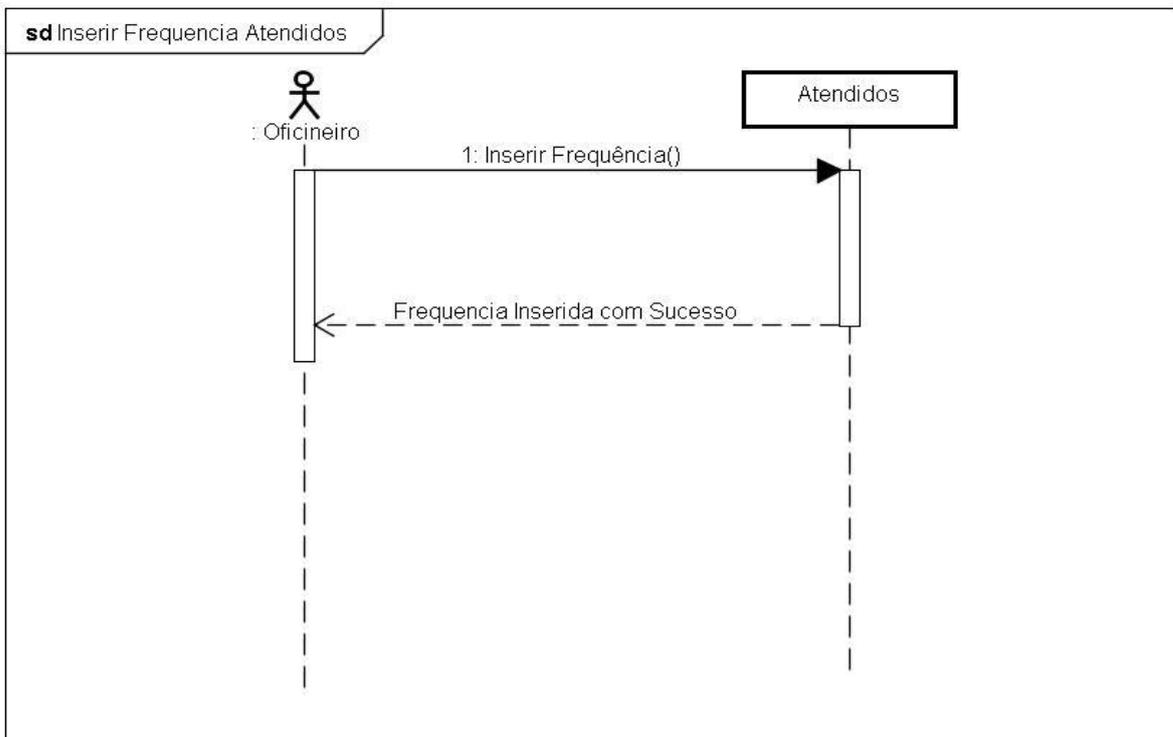


Figura 29 - Diagrama de Sequência Inserir Frequência Atendidos

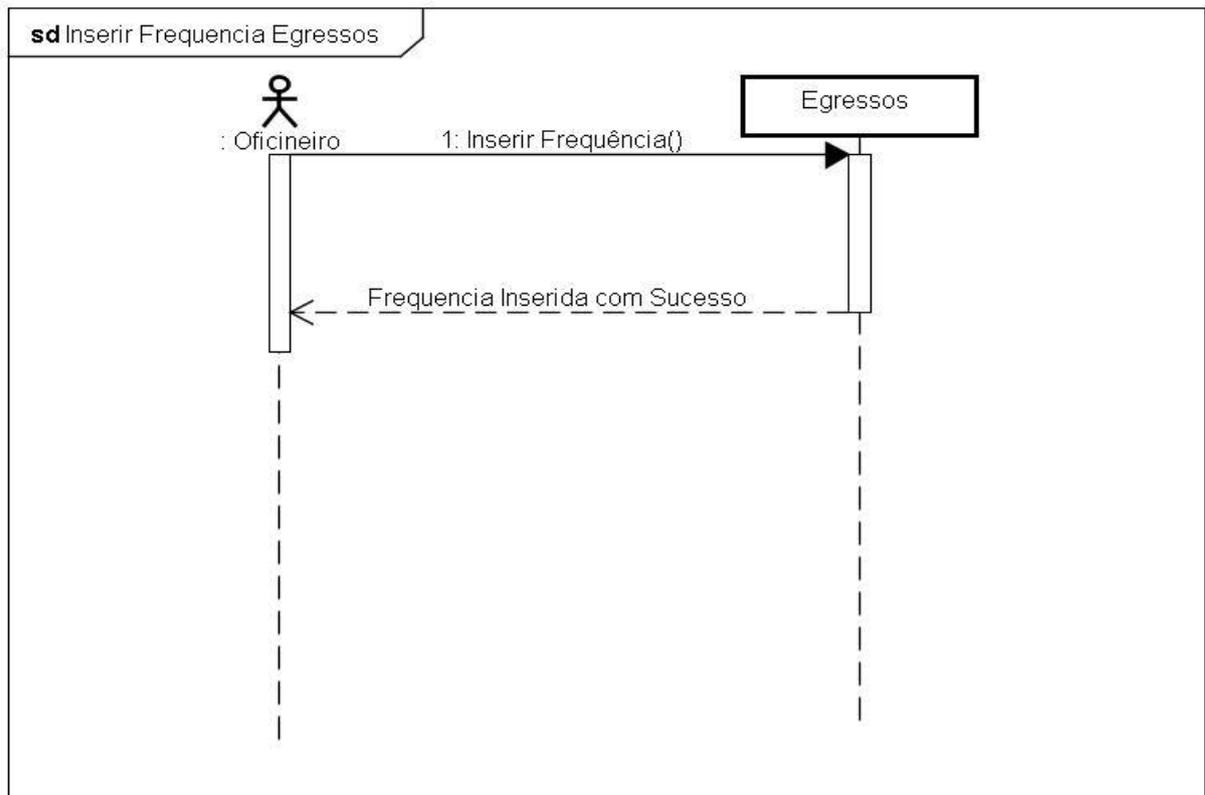


Figura 30 - Diagrama de Sequência Inserir Frequência Egressos

3.8 - DIAGRAMA DE ATIVIDADES

Um diagrama de atividade descreve como é feita a coordenação de atividades. Alguns autores consideram que os diagramas de atividade são um melhor mecanismo para mostrar dependências essenciais entre atividades realizadas por várias entidades (LEE, TEPFENHART 2001).

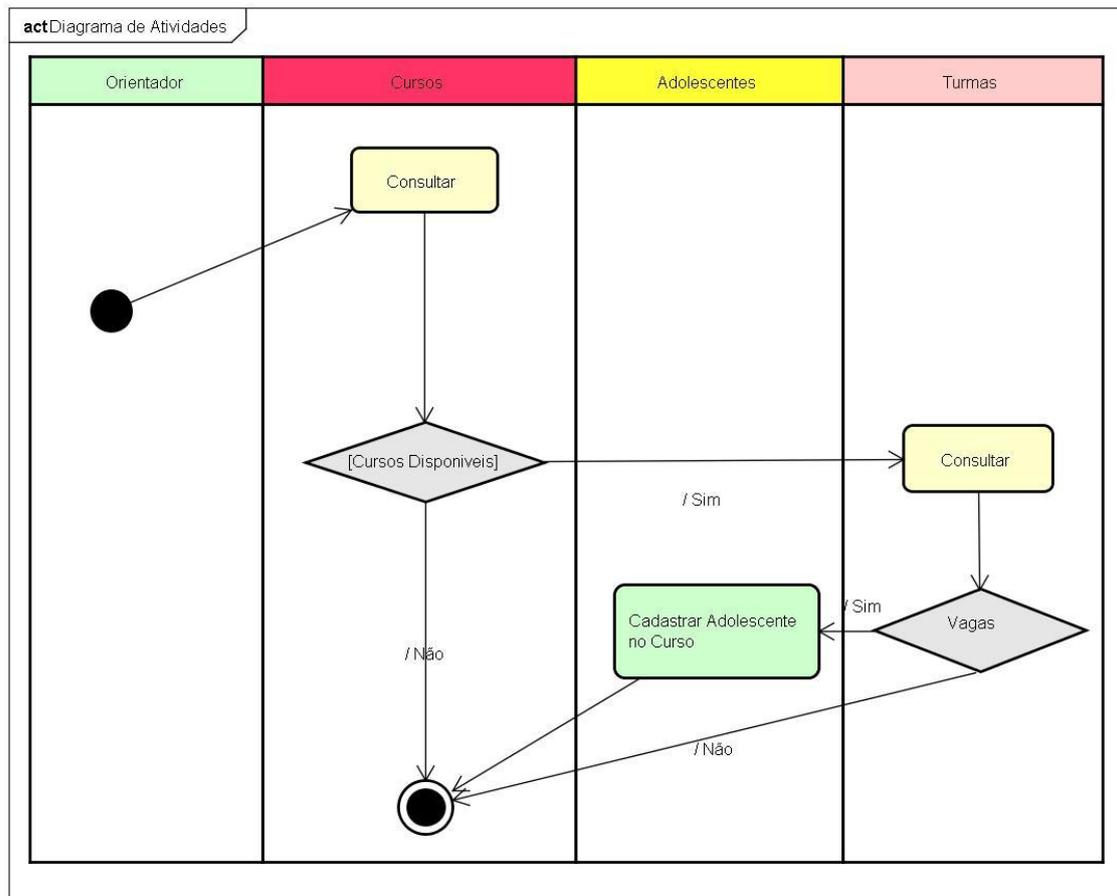


Figura 31 - Diagrama de Atividades

3.9 - DIAGRAMA ENTIDADE RELACIONAMENTO

O Diagrama Entidade Relacionamento é baseado na percepção do mundo real que consiste em um conjunto de objetos básicos chamados entidades e nos relacionamentos entre estes objetos.

Ele foi desenvolvido para facilitar o projeto de banco de dados, permitindo a especificação de um esquema de negócio, onde tal esquema representa a estrutura lógica geral do banco de dados (REZENDE, 2005).

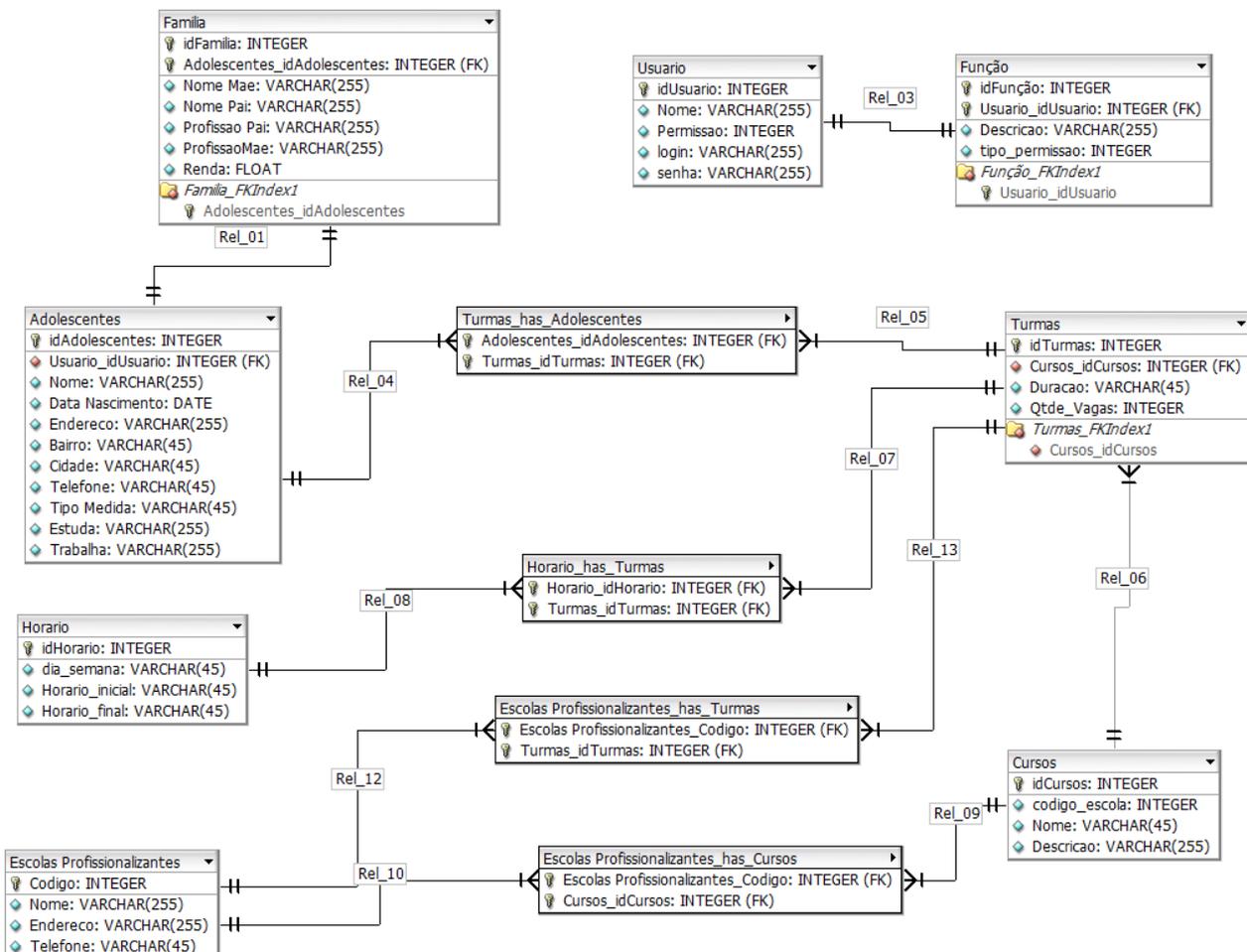


Figura 32 - Diagrama Entidade Relacionamento

4 - PROJETO

4.1 - WORK BREAKDOWN STRUCTURE

A *Work Breakdown Structure*(WBS) também conhecida como Estrutura Analítica de Trabalho é uma ferramenta de decomposição do trabalho do projeto em partes manejáveis. A WBS tem por objetivo indentificar elementos terminais (os produtos, serviços e resultados a serem feitos em um projeto). Assim, a WBS serve como base para a maior parte do planejamento de projeto. A ferramenta primária para descrever o escopo do projeto (trabalho) é a estrutura analítica do projeto WBS(<http://www.scribd.com/doc/22483180/Artigo-WBS>).

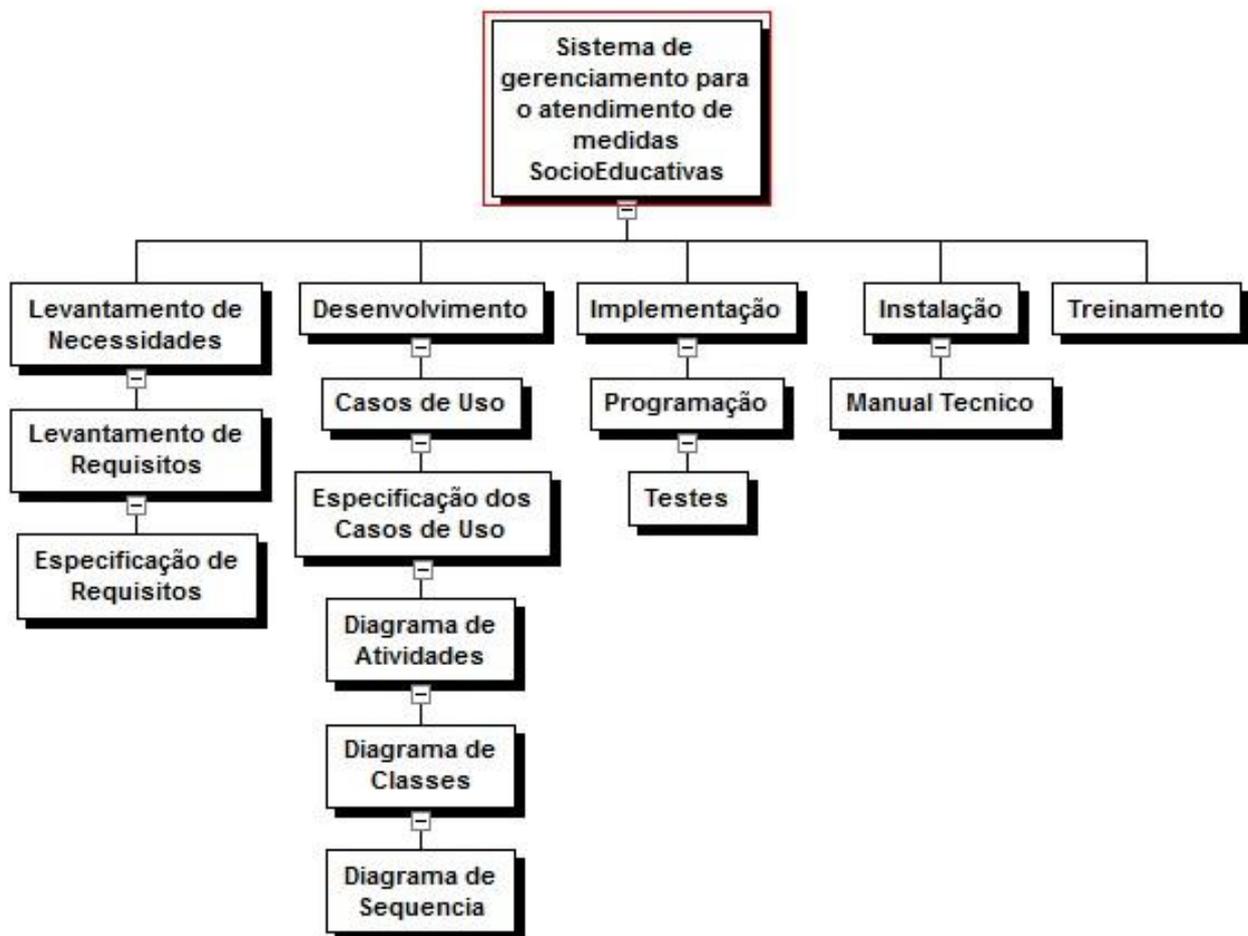


Figura 33 - Work Breakdown Structure

4.2 - SEQUENCIAMENTO DE ATIVIDADES

O diagrama abaixo mostra a sequência das atividades e o tempo de duração para a realização de cada uma delas, a serem desenvolvidas no decorrer do Projeto.

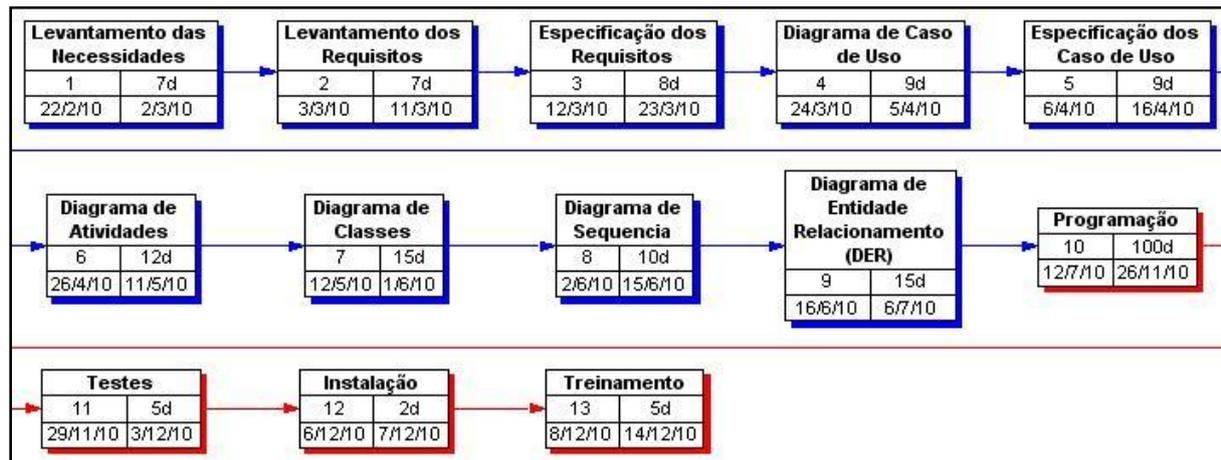


Figura 34 - Sequenciamento de Atividades

4.3 – ORÇAMENTO

4.4 – RECURSOS NECESSÁRIOS PARA O DESENVOLVIMENTO

Serão necessários para o desenvolvimento do Projeto, os recursos descritos abaixo:

- 01 Analista de Sistemas
- 01 Programador
- 01 Impressora Jato de Tinta

Analista	Quantidade Horas	Custo/hora (R\$)	Total (R\$)
Hellen	55	30,00	R\$ 1.650,00
Custo Analista			R\$ 1.650,00

Programador	Quantidade Horas	Custo/hora (R\$)	Total (R\$)
Hellen	100	20,00	R\$ 2.000,00

Total	R\$3.650,00		
--------------	--------------------	--	--

4.5 - EQUIPAMENTO

- **01 computador**

- Valor unitário = R\$2.300,00
- Dias (de uso) = 26 dias (considerar: 52h / 05 de trabalho/dia = 26)
- Depreciação = R\$2.300,00 / 12meses (01 ano. Tempo de depreciação) = R\$191,67/mês
- Custo dia = R\$191,67 / 26 (dias) = R\$7,37 (ao dia)
- Custo do computador = R\$7,37 * 52 (total de horas do projeto) = R\$383,24.

Custo do computador = 1 * R\$ 383,24 = R\$ 383,24.

- **01 impressora**

- Valor = R\$300,00
- Dias (de uso) = 26 dias
- Depreciação = R\$300,00 / 12 = R\$25,00
- Custo dia = R\$25,00 / 26 = R\$ 0,96

Custo da Impressora = R\$ 0,96 * 52 = R\$ 49,92

Custo total do Equipamento = R\$383,24 + R\$49,92 = R\$ 433,16

Custo total do Projeto = R\$ 3.650,00 + R\$ 433,16 = R\$ 4.083,16
--

5 – IMPLEMENTAÇÃO DA APLICAÇÃO

Para a implementação do sistema foi utilizado o ambiente de desenvolvimento NetBeans IDE com a linguagem de programação Java.

5.1 – ORGANIZAÇÃO DOS PACOTES E CLASSES DO SISTEMA

Para uma melhor organização, o sistema foi organizado em pacotes. A figura 35 apresenta os pacotes: pacote padrão, br.com.sistema.beans, br.com.sistema.dao, br.com.sistema.navegação, br.com.sistema.relatórios, br.com.sistema.util.



Figura 35 – Organização dos Pacotes

Pacote Padrão: Onde contém os arquivos de configuração do hibernate.

Pacote: *br.com.sistema.dao*: É nele que ficam as classes, que realizam o acesso ao banco de dados, gravam, alteram e excluem informações do banco de dados.

Pacote: *br.com.sistema.beans*: É nele que ficam as classes de modelagem do projeto, e as classes que fazem a persistência dos dados com o banco de dados.

Pacote: *br.com.sistema.navegação*: Neste pacote está definido a navegação de uma classe, é o mecanismo que permite unir algum tipo de processamento com seqüência de páginas a serem mostradas.

Pacote: *br.com.sistema.util*: Neste pacote estão adicionadas todas as classes que farão conexão com o banco de dados através do *Hibernate*.

5.2 – ORGANIZAÇÃO DAS PÁGINAS DO SISTEMA

As páginas estão divididas em diretórios, cada classe tem um diretório onde contém as páginas jsp, para cadastro e consulta, como visto abaixo temos também um diretório css e resources.

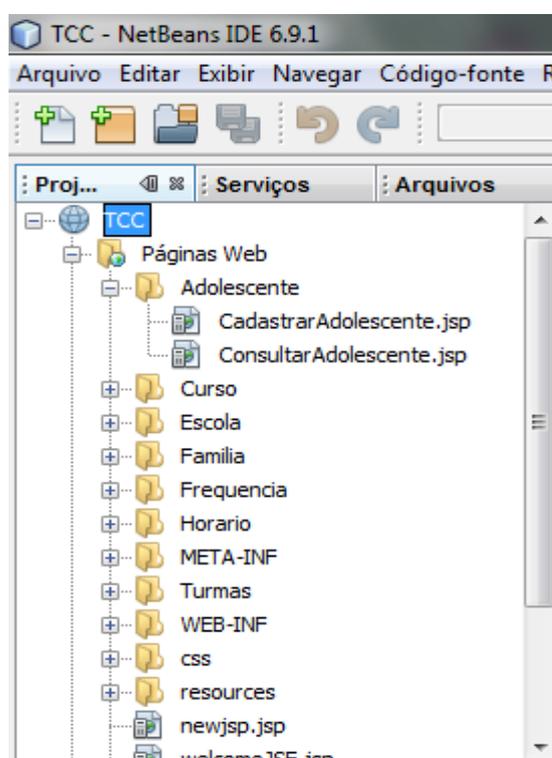


Figura 36 – Organização dos Pacotes

Diretório: *Adolescente, Curso, Escola, Família, Frequência, Horário, Turmas:* onde ficam as páginas para cadastro e consulta de cada classe.

Diretório: *css:* onde ficam os arquivos de design do projeto.

Diretório: *resources:* Onde ficam as imagens do projeto.

5.3 – INTERFACE DO SISTEMA

Ao acessar o sistema, abrirá as opções contendo: Cadastros, Movimentações, Consultas, Relatórios. Conforme a Figura 37.

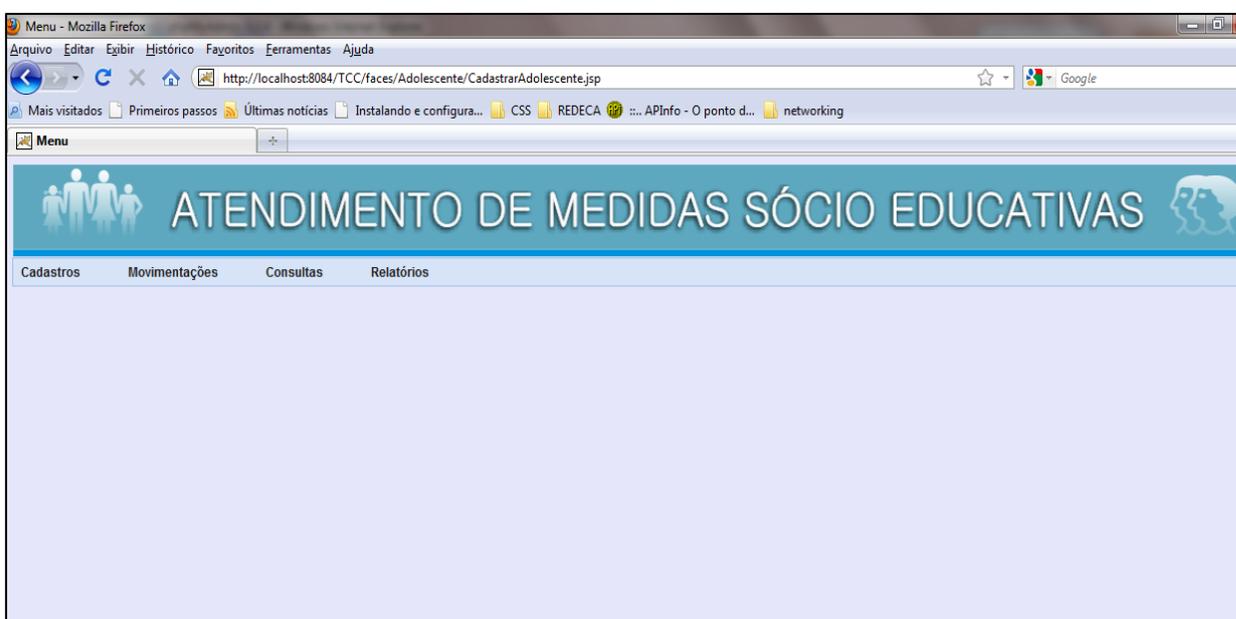


Figura 37 – Interface do Sistema

O Sistema foi desenvolvido em quatro partes: Cadastros, Movimentações, Consultas e Relatórios, este somente para emissão de relatórios.

No menu Cadastros abrirá as seguintes opções como mostra a Figura 38.



Figura 38 - Menu Cadastros

O Usuário poderá escolher uma das opções mostradas, vamos supor que a opção *Adolescente* foi escolhida. O Sistema automaticamente abrirá a janela para cadastro do adolescente, como mostra a figura 39.

The image shows a form titled 'Cadastrar Adolescentes' with the following fields and values:

Código	0
Nome	
Data Nascimento	
Endereço	
Numero	0
Bairro	
Telefone	
Tipo Medida	
Estuda	
Trabalha	

At the bottom of the form are two buttons: 'salvar' (with a green checkmark icon) and 'cancelar' (with a red X icon).

Figura 39 – Tela Cadastrar Adolescentes

Se o usuário escolher a opção cancelar o sistema voltará ao Menu Principal, se clicar em Salvar o sistema abrirá a página Consulta como mostra a figura 40, onde serão mostrados todos os adolescentes cadastrados, incluindo o que acabou de ser cadastrado.

Consulta de Adolescentes											
CODIGO	NOME	DATA NASCIMENTO	ENDEREÇO	NUMERO	BAIRRO	TELEFONE	TIPO MEDIDA	ESTUDA	TRABALHA	EDITAR	EXCLUIR
1	Hellen	03/08/1988	Estefano Paulo Ambrósio	365	CDHU	(18)-3322-0987	Liberdade Assistida	Sim	Nao		
2	Cristiane Antonio	23/07/1980	Araçatuba	598	Progresso	(18)-9706-2586	Prestação de Serviço a Comunidade	Não	Sim		
3	Marcus Felipe Pereira	19/08/1983	Senhor do Bonfim	1932	Ribeiro	(18)-3324-8456	Liberdade Assistida	Não	Sim		
4	Aline Ruisca	04/05/1982	Francisco de Campos	584	Glória	3322-52-77	Liberdade Assistida	Sim	Sim		
5	Julia Maria Artero	28/09/1989	Marília	345	Assis 4	(18) - 3322-9087	Liberdade Assistida	Não	Sim		




Figura 40 - Tela Consultar Adolescentes

Na tela Consulta o usuário poderá *Editar* e *Excluir* o Adolescente, se clicar no Editar o sistema voltará à tela de Cadastro onde o usuário poderá alterar o que quiser. A figura 41 mostrará essa opção.

Cadastrar Adolescentes

Código	5
Nome	Julia Maria Artero
Data Nascimento	28/09/1989
Endereco	Marilia
Numero	345
Bairro	Assis 4
Telefone	(18) - 3322-9087
Tipo Medida	Liberdade Assistida
Estuda	Não
Trabalha	Sim

Figura 41 - Tela Editar Adolescente

CONCLUSÃO

Com a utilização do sistema pela Entidade Social, a base de dados que compõe o mesmo pode ajudar na tomada de decisões do governo cabível, no que se diz a respeito à política pública, uma vez que na base de dados estão informações importantes sobre a frequência do atendimento sócio educativo na cidade de Assis.

A informatização das entidades sociais facilitam o atendimento à comunidade, otimizando a busca de informações na base de dados e, auxilia também, o planejamento e o rastreamento de crianças e adolescentes em conflito com a lei.

Por meio de entrevistas realizou-se todo o levantamento de requisitos e funcionalidades para o sistema e depois de realizada essa etapa, foram feitas as modelagens do sistema, como: caso de uso, diagrama de classe, e diagrama de sequência.

Os diagramas da UML foram muito importantes para o desenvolvimento do sistema, pois permitiu ter uma visão completa sobre o sistema e o que deveria ser feito para dar continuidade ao mesmo, sem a modelagem seria difícil de desenvolver a aplicação.

O conceito MVC (Modelo-Visão-Controle) também foi de suma importância para o desenvolvimento da aplicação pois este padrão permitiu separar em camadas cada uma de suas responsabilidades.

Durante o desenvolvimento do sistema, existiram dificuldades, entre elas com a linguagem de programação Java, tais dificuldades foram sanadas por meio de conversas com o orientador.

As tecnologias empregadas no trabalho mostraram - se eficientes no desenvolvimento de aplicações dinâmicas, já que essa é uma grande qualidade

da linguagem de programação Java. Tratando-se de um sistema desenvolvido para web, percebeu-se uma grande vantagem no quesito mobilidade, tal que, a aplicação poderá oferecer todas as informações sobre crianças e adolescentes atendidos pela entidade, acompanhamento de suas frequências em cursos e oficinas oferecidos pelo projeto, e irá facilitar o atendimento de medida sócio educativa, pois o mesmo era feito de forma manual.

Para desenvolvimentos futuros, pretende-se implantar o uso das informações do software Redeca (http://www.softwarepublico.gov.br/ver-comunidade?community_id=18016032), que é um software voltado para o auxílio na formação de redes sociais de proteção à criança e ao adolescente, o software não oferece informações sobre medidas sócio educativas, a junção dos dois sistemas ajudará muito os atendimentos, e o mapeamento da rede social.

REFERÊNCIAS

BOOCH, Grady; RUMBAUGH James; JACOBSON Ivar; Uml Guia do Usuário. Tradução de Fábio Freitas da Silva. Rio de Janeiro: Editora Campus, 2000.

COSTA, Daniel Gouveia; Java em Rede, recursos avançados de programação. Rio de Janeiro: Brasport, 2008.

GONÇALVES, Edson; Desenvolvendo aplicações web com Jsp, Servlets, JavaServer Faces, Hibernate, EJB 3 persistence e Ajax. Rio de Janeiro: Editora Ciência Moderna, 2007.

GONÇALVES, Edson; Desenvolvendo Aplicações Web com NetBeans IDE 6. Rio de Janeiro: Editora Ciência Moderna, 2008.

HORSTMAN, Cay S; CORNELL, Gary; Core Java 2 – Volume 1 Fundamentos. Tradução de João Eduardo Nobrega Tortello. São Paulo: Editora MAKRON Books, 2001.

JSP, Disponível em: <http://www.htmlstaff.org/ver.php?id=1592>. Acesso em 03 de Outubro de 2010.

LEE, Richard C; TEPFENHART, William M; Uml e C++ Guia Prático de Desenvolvimento Orientado a Objeto. Tradução de Celso Roberto Paschoa. São Paulo: Editora MAKRON Books, 2001.

MARTIN, James; Princípios de Análise e Projeto Baseados em Objetos. 5ª Tiragem. Rio de Janeiro: Editora Campus, 1994.

MATTOS, Érico Casella Tavares; Programação de Softwares em Java. São Paulo: Digerati Books, 2007.

MILANI, André; Mysql guia do programador. São Paulo: Editora Novatec, 2007.

REDECA, Disponível em: http://www.softwarepublico.gov.br/ver-comunidade?community_id=18016032. Acesso em 14 de Outubro de 2010.

REZENDE, Denis Alcides; Engenharia de Software e Sistema de informação. Rio de Janeiro: Brasport, 2005.

SOMERA, Guilherme; Treinamento prático em CSS. São Paulo: Digerati Books, 2006.

TALITA, Pitanga. GUJ – JavaServer Faces: A mais nova tecnologia Java para desenvolvimento WEB. São Paulo, 2009, Disponível em: <http://www.guj.com.br/JSF/tutotial/javaServerFaces.htm>.

TOMCAT, Disponível em: <http://tomcat.apache.org>. Acesso em 23 de Julho de 2010.

VARGAS, Ricardo; Gerenciamento de Projetos Estabelecendo Diferenciais Competitivos. 6º Edição. São Paulo: Editora BRASPORT Livros e Multimídias, 2005.

WBS, Disponível em: <http://www.scribd.com/doc/22483180/Artigo-WBS>. Acesso em 30 de Março de 2010.

ANEXO – CRONOGRAMA

O cronograma abaixo refere-se ao fluxo de tarefas cumpridas .

Tarefas	FEV	MAR	ABR	MAI	JUN	JUL	AGO	SET	OUT	NOV	DEZ
Levantamento de necessidades	■	■									
Levantamento de Requisitos		■									
Especificação dos Requisitos		■									
Diagrama de Casos de Uso		■	■								
Especificação de Casos de Uso			■								
Diagrama de Classes				■	■						
Diagrama de Sequência				■	■						
Diagrama de Atividades					■						
Diagrama de Implementação					■						
Diagrama de Entidade Relacionamento					■						
Elaboração da Qualificação					■						
Qualificação						■					
Programação							■	■	■	■	
Testes							■	■	■	■	■
Escrita da Versão final do Projeto								■	■	■	■
Apresentação do Projeto											■

Figura 42 - Cronograma

■ Atividades Concluídas

