



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

MAURO PEREIRA NASCIMENTO

SISTEMA PARA CLINICA DE TRATAMENTO ORTODONTICO

Assis

2014

MAURO PEREIRA NASCIMENTO

SOFTWARE PARA CLINICA DE TRATAMENTO ODONTÓLOGIA

Trabalho de conclusão de Curso
apresentado ao Instituto Municipal de
Ensino Superior de Assis, como requisito
do Curso de Análise e Desenvolvimento
de sistemas.

Orientador: Prof. Esp. Fernando Cesar Lima
Área de Concentração: Desenvolvimento de Sistemas

Assis
2014

FICHA CATALOGRÁFICA

NASCIMENTO, Mauro Pereira

Software para clinica de tratamento Ortodôntico/ Mauro Pereira Nascimento.

Fundação Educacional do Município de Assis – FEMA --

Assis, 2013.

33p.

Orientador: Prof. Esp. Fernando Cesar Lima

Trabalho de Conclusão de Curso

Instituto Municipal de Ensino Superior de Assis – IMESA.

1. Gestão Financeira.
2. Tratamento Ortodôntico

CDD: 001.61

Biblioteca da FEMA

SOFTWARE PARA CLINICA DE TRATAMENTO ORTODONTICO

MAURO PEREIRA NASCIMENTO

Trabalho de conclusão de Curso
apresentado ao Instituto Municipal de
Ensino Superior de Assis, como requisito
do Curso de Análise e Desenvolvimento
de sistemas.

Orientador: Prof. Esp. Fernando Cesar Lima

Analizador (1): Prof. Esp. Guilherme de Cleva Farto

Assis

2014

DEDICATÓRIA

Dedico esse trabalho a minha família que acreditaram no meu potencial e sempre estiveram comigo nos momentos mais difíceis.

AGRADECIMENTOS

Aos Professores da FEMA por me apoiar como professor, como orientado e como amigo nos momentos difíceis que passei. E por acreditar nos alunos que almejam conhecimento.

Minha família meus pilares! Esse trabalho é dedicado aos meus pais que sempre trabalharam com honestidade para me dar conforto e educação. E ao meu irmão Mario Soares Nascimento que me estendeu as mãos quando eu mais precisei.

O início da sabedoria é a admissão da própria ignorância. Todo o meu saber consistem em saber que nada sei.

Sócrates

RESUMO

Este sistema tem como objetivo automatizar todo processo feito na Documentação de Tratamento Ortodôntico e Ficha Financeira dos pacientes, tornando o processo mais rápido e seguro.

Com foco mais voltado para controle financeiro, que inclui, data de vencimento da mensalidade, mensalidade em atraso, descontos, acréscimo e relatórios de movimentação. Dados considerados pelo usuário como os mais importantes no sistema.

Incluindo também rotinas importantes: Agendamento de Consultas e Procedimento a Serem Realizados.

ABSTRACT

This system aims to automate the whole process done in Orthodontic Treatment Documentation and Financial Statement of the patients, making the process faster and safer.

Focusing more in financial control, including, the Maturity Date of Fee, Late Fee, Discounts, Increase and Drive Reports. Data considered by the user as the most important in the system.

Including also important routines: Scheduling Queries and Procedures to Be Achieved.

Keywords:1. Financial Management. 2. Orthodontic Treatment.

LISTA DE ILUSTRAÇÕES

Figura 1 – Diagrama Geral de Caso de Uso	21
Figura 2 - Caso de Uso Iniciar Tratamento.....	22
Figura 3 - Caso de Uso Manter Caixa	25
Figura 4 – Caso de Uso Manter Agenda	27
Figura 5 – Caso de Uso Cadastrar Usuário.....	28
Figura 6– Caso de Uso Movimentação Financeira.....	30
Figura 8 - Diagrama de Classes.....	31
Figura 9– Diagrama de Entidade Relacionamento	32

LISTA DE TABELAS

Tabela 1- Documentação do Caso de Uso Cadastrar Paciente	23
Tabela 2 - Documentação do Caso de Uso Manter Paciente.....	24
Tabela 3 - Documentação do Caso de Uso Manter Caixa	27
Tabela 4– Documentação do Caso de Uso Manter Agenda	28
Tabela 5 – Documentação do Caso de Uso Cadastrar Usuário.....	29
Tabela 6– Documentação do Caso de Uso Movimentação Financeira.....	30
Tabela 7 Cronograma.....	33

SUMÁRIO

1	INTRODUÇÃO	10
1.1	OBJETIVOS	10
1.2	PÚBLICO ALVO	11
1.3	JUSTIFICATIVAS	11
2	MÉTODO DE DESENVOLVIMENTO	12
2.1	FERRAMENTAS PARA DESENVOLVIMENTO DO SISTEMA	14
2.1.1	Java	14
2.1.1.1	Principais Características da Linguagem Java	14
2.1.1.2	Plataforma Java (JVM)	15
2.1.2	EclipseLink	16
2.1.3	Apache Tomcat	16
2.1.4	NetBeans	16
2.1.5	Jquery	17
2.2	ARMAZENAMENTO E VISUALIZAÇÃO DOS DADOS	18
2.2.1	MySQL	18
2.3	FERRAMENTAS PARA A PREPARAÇÃO DA ANÁLISE	19
2.3.1	Astah Professional	19
3	ESTRUTURA E DESENVOLVIMENTO DO SISTEMA	20
3.1	LISTA DE EVENTOS	20
3.2	LEVANTAMENTO DE REQUISITOS	20
3.3	DIAGRAMA DE CASO DE USO	21

3.3.1 Caso de Uso Iniciar Tratamento	22
3.3.1.1 Documentação do Caso de Uso Cadastrar Paciente.....	23
3.3.1.2 Documentação do Caso de Uso Manter Paciente	24
3.3.2 Caso de Uso Manter Caixa	25
3.3.2.1 Documentação do Caso de Uso Manter Caixa	26
3.3.3 Caso de Uso Manter Agenda	27
3.3.3.1 Documentação do Caso de Uso Manter Agenda.....	28
3.3.4 Caso de Uso Cadastrar Usuário	28
3.3.4.1 Documentação do Caso de Uso Cadastrar Usuário	29
3.3.5 Caso de Uso Movimentação Financeira	30
3.3.5.1 Documentação do Caso de Uso Movimentação Financeira	30
3.4 DIAGRAMA DE CLASSE	31
3.5 MODELAGEM DE ENTIDADE RELACIONAMENTO.....	32
4 CRONOGRAMA DA ESTRUTURA DE DESENVOLVIMENTO	33
5 CONCLUSÃO	34
REFERENCIAS	35

1 INTRODUÇÃO

Mudanças estão acontecendo de forma muito rápida nos mercados, tecnologias, formas organizacionais, comunicações e na forma de encarar o conhecimento, tanto tácito (que não pode ser explicitado formalmente ou facilmente transferido), quanto o codificado, que pode ser armazenado, memorizado, transacionado, transferido, reutilizado, reproduzido, comercializado a custos baixos (OLIVATTO, 2012). Desenvolver um software na área de tratamento Odontológico é uma forma de aplicar as tecnologias para usufruir dos benefícios por ela fornecidos, controlar os patrimônios da empresa e manter a integridade de dados do paciente. As clínicas de Ortodontia que não utilizam meios informatizados (*Software*) para gerenciar os seus processos, fazem todo controle manualmente ficando totalmente dependente da competência e honestidade dos colaboradores.

1.1 OBJETIVOS

O objetivo deste trabalho será a elaboração de um software que possa efetuar o controle financeiro, na organização de documentos e na automatização de todo processo de agendamento de consulta. A forma do procedimento de controle financeiro requer grande atenção, pois não poderá haver erros que possam comprometer a clínica, ou seja, o sistema não pode permitir que seja manipulado por usuários mal intencionados.

Sendo organizado dessa forma todo o setor poderá ter acesso rápido e seguro aos dados do paciente, deixando toda a responsabilidade e gerenciamento de dados para o software.

1.2 PÚBLICO ALVO

A Clínica Santana clínica que tem interesse em informatizar os processos de gestão de dados dos seus pacientes, tem seu controle totalmente manual; possui uma grande quantidade de documentos que não estão digitalizados. O controle financeiro e agendamento de consultas são feitos de uma forma que necessita da honestidade e competência da secretária.

1.3 JUSTIFICATIVAS

As clínicas de ortodontia que administram o fluxo financeiro e a forma de tratamento ortodôntico através de documentos físicos (papel), tem a maior possibilidade de cometer erros como perda de documentos e cobranças indevidas.

Com o avanço da tecnologia e de inúmeros softwares no mercado as clínicas não podem estar sujeitas a erros como os citados acima. Porém existe uma grande dificuldade de encontrar um software que seja adequado e supra as necessidades da clínica. Problema que ocorre pelo fato de que o software tem que se adequar a forma que a clínica trabalha não o inverso.

Por esses motivos entre outro como falta de suporte, as clínicas que abrangem essa área de tratamento ortodôntico encontram dificuldade em empregar o uso da tecnologia.

2 MÉTODO DE DESENVOLVIMENTO

Para desenvolvimento desse trabalho será utilizado as mais novas tecnologia para desenvolvimento de aplicativo web. A linguagem JAVA, uma linguagem de alto nível que possibilita o desenvolvimento de software de qualidade, como utilitários, jogos e aplicativos corporativos, entre muitos outros.

NetBeans é uma ferramenta para desenvolvimento que auxilia na implementação, permite integrar (*Framework*) ferramentas desenvolvidas para serem utilizadas de forma auxiliar ou integrada no desenvolvimento de *software*.

A arquitetura do projeto faz uso do padrão MVC, um padrão utilizado para separar por partes cada etapa de desenvolvimento, facilitando possíveis manutenções.

Um *Framework* representa uma aplicação reutilizável e semicompleta que pode ser implantada em aplicações personalizadas.

Tem como vantagem a redução de esforço em programação, pois o *framework* oferece um conjunto de algoritmos eficientes, prontos para serem utilizados e reutilizados no desenvolvimento.

EclipseLink é um exemplo de *framework* que realiza mapeamento de objeto-relacional de forma automática. Oferece mais facilidade ao programador nos momentos de implementação de códigos para gravar, alterar ou remover dados do banco de dados, sem a necessidade de digitar complexos códigos SQL.

Tomcat da Apache atua como servido *web*, muito utilizado, pois é leve e de fácil compreensão.

A utilização de *framework* para mapeamento automático de objeto-relacional de banco de dados, trás alguns incômodos quando for necessário migrar de *framework*, ou seja, efetuar alterações no código-fonte. A fim de evitar esse incomodo foi desenvolvido a especificação JPA. A especificação JPA representa um padrão que permite que seu objeto seja gravado diretamente em um banco de dados relacional, sem a necessidade de utilização direta da API JDBC. O mapeamento objeto relacional ocorre entre os objetos Java e as tabelas de um banco de dados relacional.

Jquery uma biblioteca *javascript cross-browser* muito poderosa, sendo integrada a grandes ferramentas de desenvolvimento web como visual studio 2012 da Microsoft.

Armazenamento de dados, MySQL é um sistema de gerenciamento de bancos de dados relacional. Foi desenvolvido originalmente para lidar com bancos de dados muito grandes de maneira muito mais rápida que as soluções existentes e tem sido usado em ambientes de produção de alta demanda. A conectividade, velocidade, e segurança fazem com que o MySQL seja altamente adaptável para acessar bancos de dados na Internet.

2.1 FERRAMENTAS PARA DESENVOLVIMENTO DO SISTEMA

2.1.1 Java

Lançada pela Sun Microsystems em 1995. Java é uma linguagem de programação orientada a objetos que começou a ser criada em 1991, em San Hill Road empresa filiada a Sun (da qual hoje pertence a empresa Oracle), teve início com o Green Project, no qual os mentores foram Patrick Naughton, Mike Sheridan, e James Gosling. No início não tinha intenção de criar uma linguagem de programação, mais sim de antecipar as inovações que eram previstas nas áreas de informática e programação.

Nesta época, a internet estava ficando cada vez mais popular, e a equipe do Green Project começou a pensar em aplicações. Eles conseguiram adaptar a linguagem Oak para a internet, e em 1995 foi lançado o Java, que era uma versão atualizada do Oak para a internet. A tecnologia Java teve uma enorme utilização, e logo começou a ser utilizado pelas grandes empresas como a IBM. Java é uma linguagem relativamente simples e dinâmica, permite criar programas e aplicações para a Web sem depender de outra linguagem. (PACIEVITCH).

2.1.1.1 Principais Características da Linguagem Java

A linguagem Java foi projetada tendo em vista os seguintes objetivos:

- Orientação a objeto - Baseado no modelo de *Smalltalk* o pioneirismo de programação orientada a objetos.
- Portabilidade - Independência de plataforma uma característica que torna o aplicativo multi-plataforma pelo fato de ser interpretado pela máquina virtual;
- Recursos de Rede - Possui extensa biblioteca de rotinas que facilitam a cooperação com protocolos TCP / IP, como HTTP e FTP. Exemplo criar um chat de mensagem de texto entre máquinas através de Socket e SocketServer;

Além disso, podem-se destacar outras vantagens apresentadas pela linguagem:

- Sintaxe similar a Linguagem C / C + +;

- Facilidades de Internacionalização - Suporta nativamente caracteres Unicode um padrão que permite aos computadores representar e manipular, de forma consistente, texto de qualquer sistema de escrita existente;
- Simplicidade na especificação, tanto da linguagem como do “ambiente” de execução (JVM);
- É distribuída com um vasto conjunto de bibliotecas (ou APIs);
- Libera espaço de memória automática *Garbage Collector* (Lixeiro) ao ser executado elimina objetos que não tem referências;
- Carga Dinâmica de Código - Programas em Java são formados por uma coleção de classes armazenadas independentemente e que podem ser carregadas no momento de utilização (PALMEIRA).

2.1.1.2 Plataforma Java (JVM)

Java Virtual Machine é uma máquina de computação abstrata. Como uma máquina de computador real pode manipular várias áreas de memória em tempo de execução. É a peça chave para fornecer capacidade de multiplataforma para as aplicações Java. A JVM é a máquina virtual responsável por interpretar e executar o código Java compilado (bytecode) e, portanto são provedoras de formas e meios de que o aplicativo possa conversar com o sistema operacional.

Esta abstração viabiliza as implementações da JVM para diferentes plataformas de hardware e de sistemas operacionais, o que possibilita que aplicativos Java sejam multiplataforma (PALMEIRA).

2.1.2 EclipseLink

EclipseLink é baseado no TopLink que é um *framework* para mapeamento de objeto-relacional de banco de dados desenvolvido pela Oracle;

Na qual contribuiu com o código-fonte para criar o projeto EclipseLink. Projeto da Eclipse Foundation, que possui código fonte aberto que possibilita interagirem com vários tipos de serviços de informação como Banco de Dados, Web Services, Objetos XML, EIS, etc. Sendo assim, EclipseLink não implementa apenas o padrão JPA, mas também outros padrões como JAXB, JCA e SDO (MEDEIROS).

2.1.3 Apache Tomcat

Desenvolvido pela Fundação Apache e distribuído como *software* livre sejam para fins comerciais ou não. Tomcat é um *software* que permite a execução de aplicações baseadas em Servlet, JSP (Java Server Pages), JSTL (Java Standard Tag Library), JSF (Java Server Faces) e outras tecnologias para aplicações web. O servidor Tomcat trás junto consigo uma aplicação web chamada Tomcat Manager. Esta aplicação web é instalada no contexto/manager e através de qualquer *browser* provê funcionalidades básicas para gerenciar as aplicações web que estão sendo executada no servidor (MEDEIROS).

2.1.4 NetBeans

O NetBeans começou como um projeto estudantil (originalmente denominado Xelfi) na República Tcheca em 1996. O objetivo era escrever um Java IDE similar ao Delphi no Java. Roman Stanek, um empreendedador que já tinha se envolvido em diversas empresas embrionárias na República Tcheca. Ele estava procurando por uma boa idéia na qual investir e descobriu o Xelfi. Em 1999, o NetBeans DeveloperX2 foi lançado, suportando o Swing. No mesmo ano Net Beans tornou uma ferramenta de desenvolvimento viável graças ao desempenho do JDK 1.3. No mesmo ano a Sun Microsystems adquiriu a ferramenta NetBeans e uma outra

empresa de ferramentas, a Forté, ao mesmo tempo, e decidiu renomear o NetBeans como *Forté para Java*. Movendo rápido adiante menos de seis meses depois, foi tomada decisão que o NetBeans seria de código-fonte aberto. Melhor suporte para Tecnologias Java Mais Recentes: O NetBeans IDE oferece suporte abrangente e de primeira classe para as tecnologias e aprimoramentos Java mais recentes, antes de outros IDEs é a primeiro IDE a oferecer suporte a JDK 7, Java EE 6 e JavaFX 2.

Edição de Códigos Rápida & Inteligente: O Editor do NetBeans recua linhas, associa palavras e colchetes e realça códigos-fonte sintática e semanticamente. Ele também fornece modelos de código, dicas de codificação e ferramentas de refatoração. O editor suporta várias linguagens, incluindo Java, C/C++, XML, HTML, PHP, Groovy, Javadoc, JavaScript e JSP. Como o editor é extensível, você pode adicionar suporte para muitas outras linguagens.

Escreva Códigos sem Erros: para a identificação e correção de problemas comuns em código Java, NetBeans oferece uma ferramentas de análise estática com integração especial com a ferramenta FindBugs. Atualmente o NetBeans IDE encontra-se na versão 7.3, disponível para download gratuito no site netbeans.org (FOX, 2001);

2.1.5 JQuery

No dia 22 de agosto de 2005 John Resig desenvolvedor americano profundo conhecedor de JavaScript atuando na Corporação Mozilla e autor do livro *Pro JavaScript Techniques* escreveu em seu blog um artigo relatando sua frustração com a maneira verbosa de se escrever JavaScript para obter os resultados pretendidos. Naquele artigo publicou alguns exemplos no quais propunha o uso de seletores CSS com o objetivo de simplificar e dar maior versatilidade ao código. Escreveu, então, que aquela ainda não era uma forma definitiva do que tinha em mente, mas iria aperfeiçoar e testar suas propostas. O nome ainda não existia mais ali foi lançada a idéia que resultaria em jQuery. Aproximadamente cinco meses após a publicação do artigo em seu blog John Resig apresenta publicamente os

resultados de seus estudos no BarCampNYC - Wrap up realizado no dia 14 de janeiro de 2006 em uma palestra sob o título "jQuery a nova onda para JavaScript".

O ano de 2006 marcou ainda a criação do primeiro plug-in, o lançamento de uma versão não obstrutiva de LightBox usando a biblioteca jQuery, o lançamento das versões 1.0, 1.0.1, 1.0.2, 1.0.3 e 1.04, o lançamento da versão XML da biblioteca e o primeiro conteste público de criação com jQuery.

jQuery é uma biblioteca JavaScript disponibilizada como software livre e aberto, ou seja, de emprego e uso regido segundo licença conforme as regras estabelecidas pelo MIT – Massachusetts *Institute of Technology* ou pelo GPL – GNU *General Public License*. Isso, resumidamente, significa que você pode usar a biblioteca gratuitamente tanto em desenvolvimento de projetos pessoais como comerciais. JavaScript não é uma linguagem java, apenas foi dado o nome por *Marketing*. Criada para facilitar efeitos em sites sem a necessidade de escrever longas linhas de código. uma maneira simples e fácil de escrever JavaScript colocada ao alcance não só de programadores experientes como também de designers e desenvolvedores com pouco conhecimento de programação (SILVA, 2008).

2.2 ARMAZENAMENTO E VISUALIZAÇÃO DOS DADOS

Local onde os dados são armazenados para que possam ser consultados quando necessário.

2.2.1 MySQL

Desenvolvido pela empresa sueca MySQL AB e publicado, originalmente, em maio de 1995. MySQL é um sistema gerenciador de banco de dados relacional de código aberto usado na maioria das aplicações gratuitas para gerir suas bases de dados. Após, a empresa foi comprada pela Sun Microsystems e, em janeiro de 2010, integrou a transação bilionária da compra da Sun pela Oracle Corporation. Atualmente, a Oracle (PISA, 2012). MySQL é um banco de dados relacional, desenvolvido para plataformas Linux-like, OS/2, Windows. Muito utilizado para servidor web.

Compatível com o padrão SQL linguagem essa amplamente utilizada para manipulação de dados sendo considerada uma ferramenta de manipulação de base de dados de tamanho moderado. Tem como qualidade sua velocidade proporcionada pela sua implementação leve que não inclui na totalidade o suporte as instruções SQL, é um banco de dados multiprocessados, significando que pode utilizar vários processadores ao mesmo tempo, Possui API's para C, C++, Java, Perl, PHP, Python e TCL, permite conexões via TCP/IP e Permite acesso via ODBC. (REDAÇÃO, 2007).

2.3 FERRAMENTAS PARA A PREPARAÇÃO DA ANÁLISE

Software que serão utilizado para desenvolver o UML (Unified Modeling Language). Uma linguagem que define em forma de diagramas como será desenvolvido o projeto.

2.3.1 **Astah Professional**

A UML que significa Linguagem Unificada de Modelagem é uma linguagem padrão para modelagem orientada a objetos. Não é um método de desenvolvimento. Têm como papel auxiliar a visualizar o desenho e a comunicação entre objetos. Ela permite que desenvolvedores visualizem os produtos de seu trabalho em diagramas padronizados, e é muito usada para criar modelos de sistemas de software. Utiliza-se de um conjunto de técnicas de notação gráfica para criar modelos visuais de software de sistemas intensivos, combinando as melhores técnicas de modelagem de dados, negócios, objetos e componentes. É uma linguagem de modelagem única, comum e amplamente utilizável.

Para criar esses diagramas necessitamos de ferramentas específicas para substituir o papel e a caneta. Um dos softwares mais conceituados para a criação de diagramas de UML é o Astah.

Astah é um software pago, mas possui uma versão gratuita, porem alguns recursos não estão liberados para essas versões gratuitas. Uma ferramenta intuitiva e de fácil utilização.

3 ESTRUTURA E DESENVOLVIMENTO DO SISTEMA

3.1 LISTA DE EVENTOS

Início de Tratamento evento que ocorre quando o paciente inicia o tratamento. Manter Caixa será o fluxo de caixa que ocorrerão durante o dia. Receber Mensalidade quando paciente efetua o pagamento da parcela.

Parcelas em Atraso são recebimentos não realizados. Parar o Tratamento ocorre quando o paciente para de realizar manutenções no aparelho ortodôntico. Retorno ao Tratamento ocorre quando usuário retorna a efetuar o tratamento parado. Abandono de Tratamento paciente que não retorna ao tratamento e não apresenta justificativa. Agendar Consulta ocorre quando o paciente solicita agendamento de consulta. Relatório Financeiro é informado de acordo com a solicitação do usuário. Relatório de Pacientes emite uma lista de paciente da clinica. Procedimentos Realizados um prontuário com todos os procedimentos realizados no tratamento do paciente.

3.2 LEVANTAMENTO DE REQUISITOS

O levantamento de requisitos foi passado pela Dr^a Leda Santana proprietária da clinica de tratamento ortodôntico Clinica Santana, que procurava uma forma de informatizar a sua clinica. Na análise foram recolhidas as informações sobre os processos em que a clinica era regida. Com o decorrer da análise os dados relevantes obtidos, foram, desde a procura do paciente para realizar um orçamento, a forma que o orçamento era cadastrado e sua validade. O processo percorrido após a validação do orçamento. A forma que era gerada as parcela. No fluxo de caixa quem poderia realizar o recebimento dessas parcelas e como eventualmente poderia ser realizada ou paga. A documentação gerada no decorrer do tratamento do paciente e qual procedimento seria adotado caso ocorra a desistência do tratamento.

3.3 DIAGRAMA DE CASO DE USO

A figura abaixo descreve o sistema do ponto de vista do usuário.

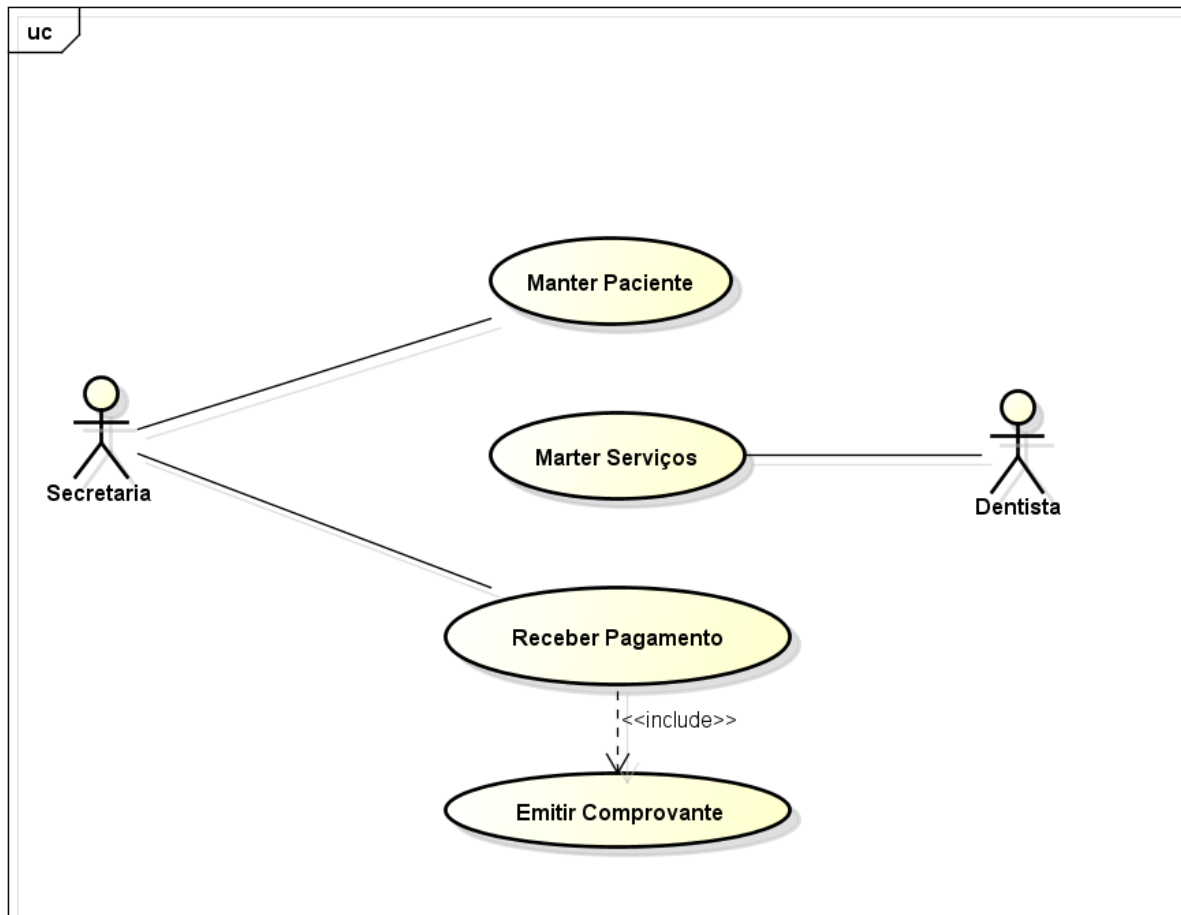


Figura 1 – Diagrama Geral de Caso de Uso

3.3.1 Caso de Uso Iniciar Tratamento

Afigura abaixo ilustra o cadastro de cliente

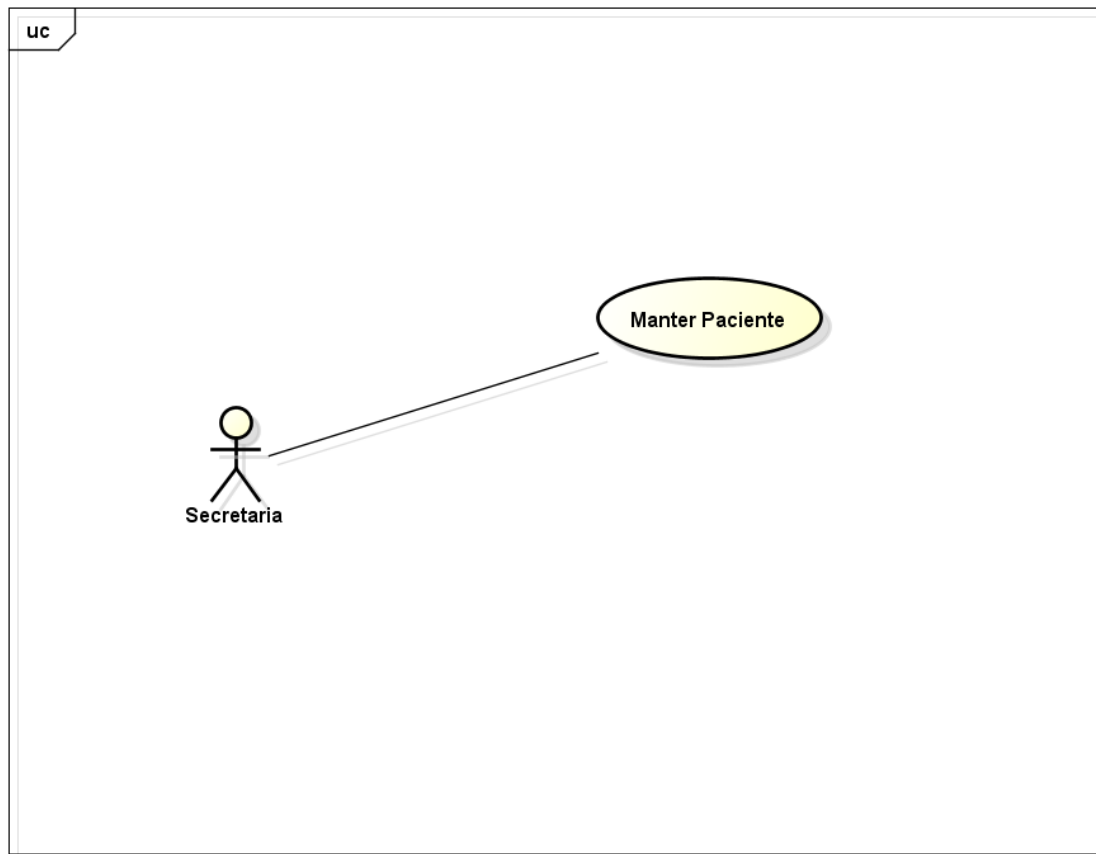


Figura 2 - Caso de Uso Iniciar Tratamento

3.3.1.1 Documentação do Caso de Uso Cadastrar Paciente

Nome do Caso de Uso	Cadastrar Paciente
Caso de Uso Geral	
Ator Principal	Dentista
Ator Secundário	Secretaria
Resumo	Este caso de uso descreve as etapas percorridas para efetuar cadastro de paciente.
Pré-Condições	Ator valido.
Pós-Condições	
Fluxo Principal	
Ação de Ator	Ação do Sistema
1. Solicitar CPF ou RG do paciente.	
	2. Identifica o caso de uso Cadastrar Paciente.
	3. Se necessário executar caso de uso Manter Paciente. Caso já houver paciente cadastrado com o mesmo numero de CPF ou RG.
4. Informa os dados necessários para cadastro de paciente.	
	5. Se solicitado grava informações.
Restrições/Validações	1. O CPF ou RG precisam ser validos
	2. Preencher todos os campos Obrigatórios.

Tabela 1- Documentação do Caso de Uso Cadastrar Paciente

3.3.1.2 Documentação do Caso de Uso Manter Paciente

Nome do Caso de Uso	Manter Paciente
Caso de Uso Geral	Cadastrar Paciente
Ator Principal	Dentista
Ator Secundário	Secretaria
Resumo	Descreve as possíveis atividades de manutenção do Cadastro de Paciente, ou seja, permite incluir, alterar ou consultar dados do paciente. Eventos Parcelas em Atraso Parar o Tratamento Retorno ao Tratamento Abandono de Tratamento estão incluídos
Pré-Condições	Ator valido.
Pós-Condições	
Fluxo Principal	
Ação de Ator	Ação do Sistema
1. Solicitar CPF ou RG do paciente.	
	2. Ao informar o CPF ou RG automaticamente será verificado se existe cadastrado com o CPF ou RG informados, apresentar seus dados.
3. Se necessário, alterar ou inserir os dados do paciente	
	4. Gravar as atualizações
Restrições/Validações	5. O CPF ou RG precisam ser validos
	6. Preencher todos os campos Obrigatórios.
Fluxo Exceção – Histórico do Paciente	
Ação de Ator	Ação do Sistema
1. Seleciona o Paciente	
	2. Informa os dados do paciente
3. Solicita histórico do paciente	
	4. Informa o histórico médico, tratamento realizado até ultima data de consulta, pagamento e pendência de mensalidades.

Tabela 2 - Documentação do Caso de Uso Manter Paciente

3.3.2 Caso de Uso Manter Caixa

A figura abaixo descreve a movimentação de caixa

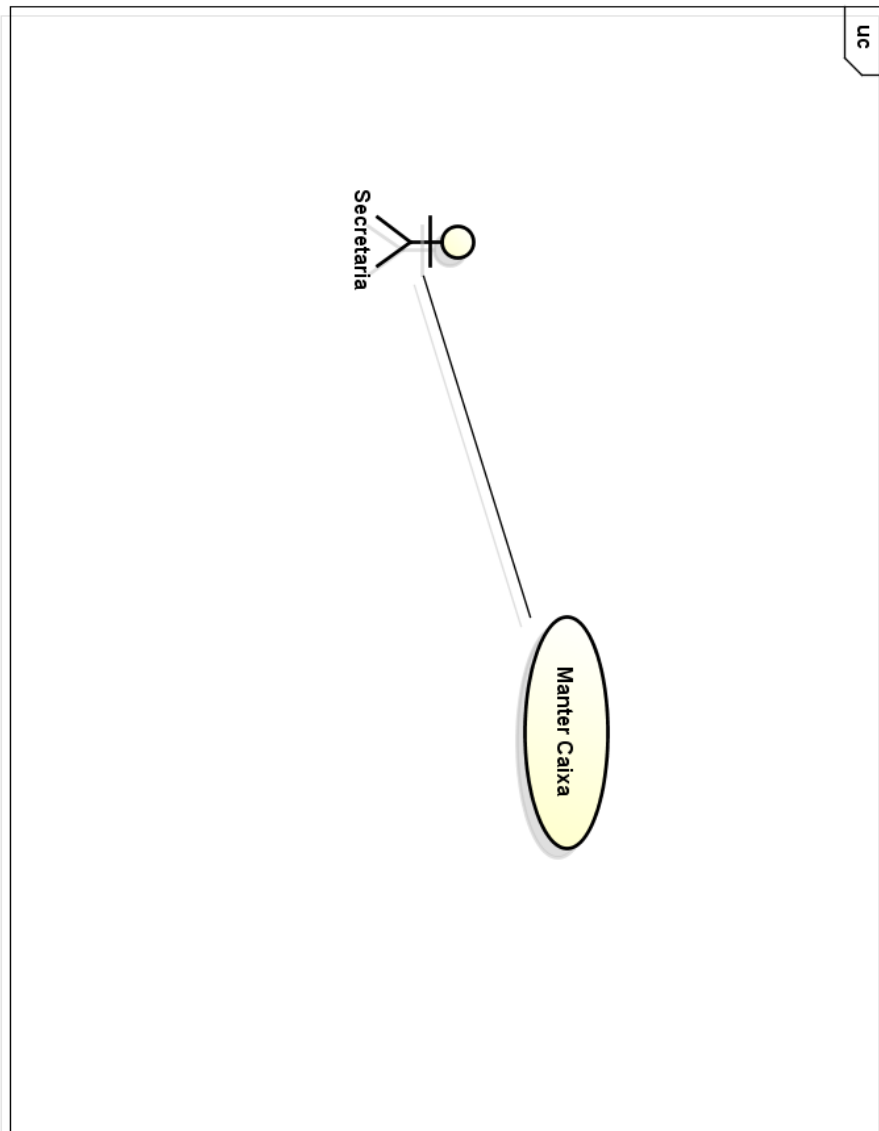


Figura 3 - Caso de Uso Manter Caixa

3.3.2.1 Documentação do Caso de Uso Manter Caixa

Nome do Caso de Uso	Manter Caixa
Caso de Uso Geral	Manter Caixa
Ator Principal	Dentista
Ator Secundário	Secretaria
Resumo	Este caso de uso descreve as possíveis atividades de manutenção no caixa, permite receber, pagar, abrir caixa e fechar caixa.
Pré-Condições	Ator valido / Caixa fechado
Pós-Condições	O caixa necessita ser fechado
Fluxo Principal	
Ação de Ator	Ação do Sistema
1. Abrir caixa	
	2. Solicita valor em dinheiro
3. Informa o valor	
	4. Informa caixa aberto
Restrições/Validações	A abertura de caixa somente pode ser feita com valor em dinheiro
Fluxo Alternativo – Receber Mensalidade	
Ação de Ator	Ação do Sistema
1. Seleciona Receber	
	2. Solicita nome do paciente
3. Informa o nome	
	4. Consultar mensalidade a ser paga
	5. Informar valores acréscimo
	6. Calcular valor total da mensalidade
	7. Apresentar valor da mensalidade
8. Seleciona a mensalidade a pagar e Realizar pagamento	
	9. Solicita a forma de pagamento
10. Seleciona a forma de pagamento	
11. Seleciona confirmar pagamento	
	12. Altera o status da mensalidade para recebido.
Restrições/Validações	1. Somente poderá ser alterado pelo ator principal 2. Emite comprovante (recibo) 3. Registra o pagamento no histórico financeiro do paciente
Fluxo Alternativo – Efetuar Pagamento	
Ação de Ator	Ação do Sistema
1. Seleciona Efetuar Pagamento	

	2. Solicita valor
3. Informa o valor	
	4. Verifica valor disponível em caixa
	5. Solicita a origem do pagamento
Informa os dados	
	6. Efetua o pagamento
Restrições/Validações	<ol style="list-style-type: none"> 1. O pagamento poderá ser realizado se tiver dinheiro em caixa suficiente. 2. Registra a saída de caixa no histórico de movimentação de caixa

Tabela 3 - Documentação do Caso de Uso Manter Caixa

3.3.3 Caso de Uso Manter Agenda

A figura abaixo descreve a forma que ocorrerá o agendamento.

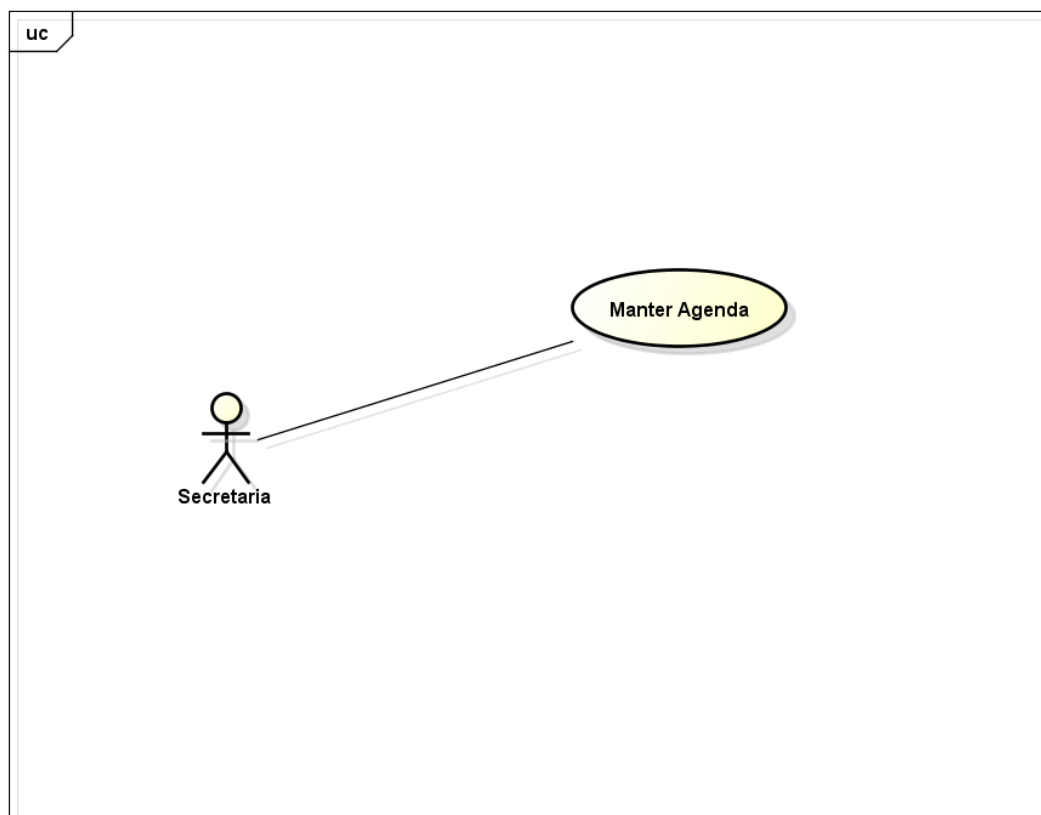


Figura 4 – Caso de Uso Manter Agenda

3.3.3.1 Documentação do Caso de Uso Manter Agenda

Nome do Caso de Uso	Manter Agenda
Caso de Uso Geral	
Ator Principal	Dentista
Ator Secundário	Secretaria
Resumo	Este caso de uso descreve as etapas percorridas para efetuar agendamento de consultas e possíveis alterações.
Pré-Condições	Paciente cadastrado
Pós-Condições	Data da consulta marcada.
Fluxo Principal	
Ação de Ator	Ação do Sistema
1. Solicitar CPF ou RG do paciente.	
	2. Verifica se existe cadastro com os dados fornecido.
	Disponibiliza as datas disponível para agendar consulta.
3. Informa a data da consulta	
	4. Registra consulta
Restrições/Validações	1. Data deve estar disponível para ser marcada

Tabela 4– Documentação do Caso de Uso Manter Agenda

3.3.4 Caso de Uso Cadastrar Usuário

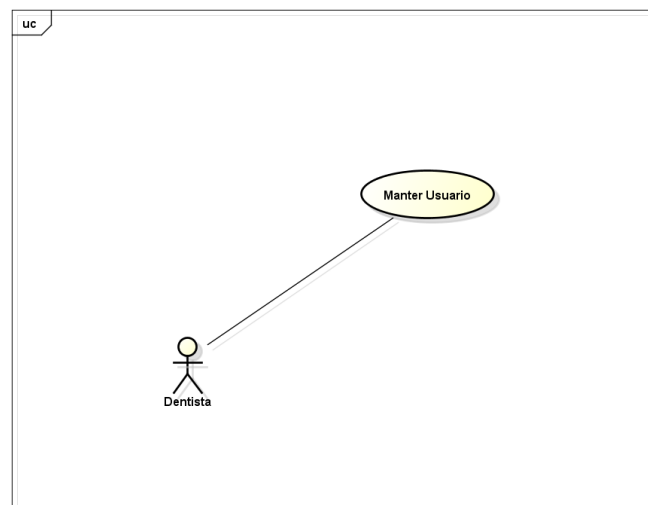


Figura 5 – Caso de Uso Cadastrar Usuário

3.3.4.1 Documentação do Caso de Uso Cadastrar Usuário

Nome do Caso de Uso	Cadastrar Usuário
Caso de Uso Geral	
Ator Principal	Dentista
Ator Secundário	
Resumo	Este caso de uso descreve as etapas percorridas para efetuar cadastro de Usuário.
Pré-Condições	Ator valido.
Pós-Condições	
Fluxo Principal	
Ação de Ator	Ação do Sistema
1. Solicitar CPF ou RG do Usuário.	
	2. Identifica o caso de uso Cadastrar Usuário.
3. Informa os dados necessários para cadastro de usuário.	
	4. Se solicitado gravar informações.
Restrições/Validações	1. O CPF ou RG precisam ser validos
	2. Preencher todos os campos Obrigatórios.
Fluxo Alternativo – Manter Usuário	
Ação de Ator	Ação do Sistema
	Mostra o cadastro do usuário
Efetua a manutenção	
	Se solicitado atualizar o cadastro
Restrições/Validações	

Tabela 5 – Documentação do Caso de Uso Cadastrar Usuário

3.3.5 Caso de Uso Movimentação de Caixa

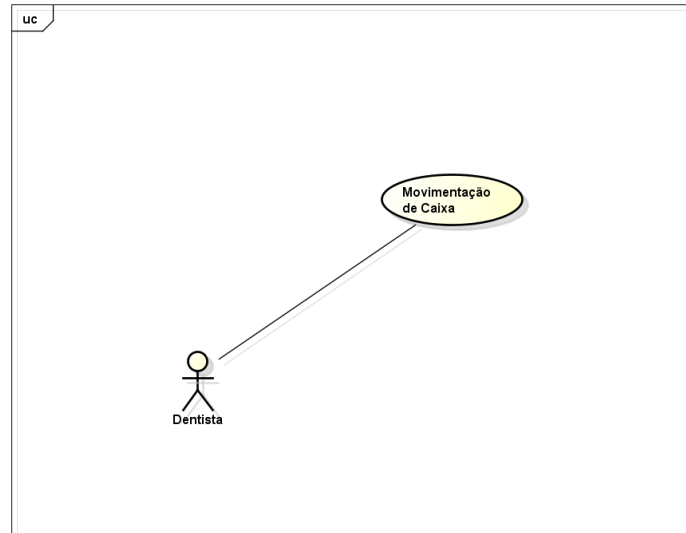


Figura 6– Caso de Uso Movimentação de Caixa

3.3.5.1 Documentação do Caso de Uso Movimentação Financeira

Nome do Caso de Uso	Movimentação de Caixa
Caso de Uso Geral	
Ator Principal	Dentista
Ator Secundário	
Resumo	Este caso de uso mostra o DRE do período solicitado.
Pré-Condições	Ator valido.
Pós-Condições	
Fluxo Principal	
Ação de Ator	Ação do Sistema
1. Seleciona o período	
	Lista todas as movimentações diárias realizadas no período solicitado. Evento Relatório Financeiro
	2. Se solicitado imprimir relatório.
Restrições/Validações	1. O período final deve ser o dia anterior da data atual.

Tabela 6– Documentação do Caso de Uso Movimentação de Caixa

3.4 DIAGRAMA DE CLASSE

Diagrama de Classe possui atributos e métodos mapeados de acordo com o DER (Diagrama Entidade de Relacionamento). Conforme ilustrado na figura abaixo.

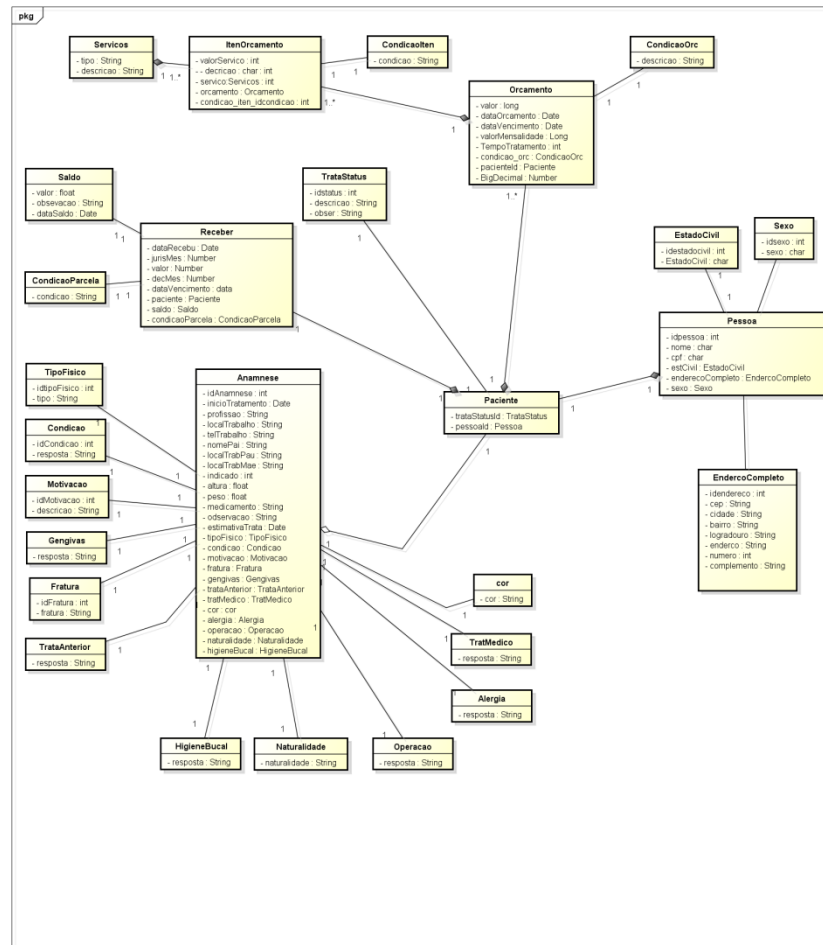


Figura 7 - Diagrama de Classes

3.5 MODELAGEM DE ENTIDADE RELACIONAMENTO

A figura abaixo ilustra a modelagem de tabelas relacionada para o banco de dados.

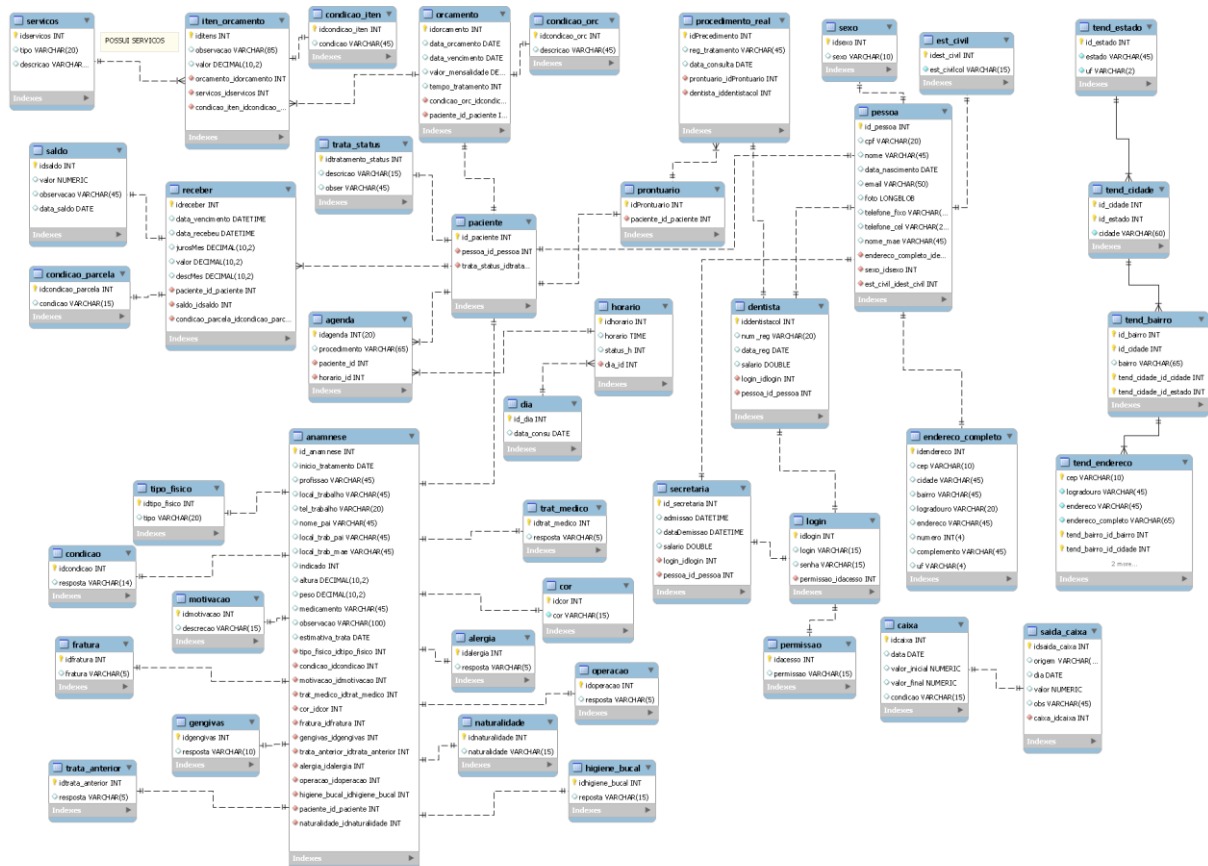


Figura 8– Diagrama de Entidade Relacionamento

4 CRONOGRAMA DA ESTRUTURA DE DESENVOLVIMENTO

	FEV	MAR	ABR	MAI	JUN	JUL	AGO	SET	OUT	NOV	DEZ
Levantamento de Requisitos	■										
Especificação de Requisitos		■	■	■	■						
Validação de Requisitos		■	■	■							
Caso de Uso		■	■	■	■	■					
Narrativa de Caso de Uso		■	■	■	■	■					
Escrita Exame Qualificação			■	■	■	■					
Defesa da Qualificação						■					
Escrita da Versão Final							■	■	■	■	
Diagrama de Atividades			■	■	■	■					
Diagrama de Classes				■	■	■					
Diagrama de Seqüência				■	■	■					
Diagrama ER					■	■					
Programação						■	■	■	■	■	■
Teste							■	■	■	■	■
Implantação/Validação											■
Defesa do TCC											■

Tabela 7 Cronograma

5 CONCLUSÃO

O desenvolvimento do software específico para clínica de tratamento ortodôntico desenvolvido neste trabalho.

Seu ambiente de execução será em navegadores (Browser) de internet, tornando-se acessível de qualquer dispositivo que possua acesso à internet.

Organiza toda a documentação e disponibiliza as informações armazenada de forma rápida e segura .

Possui interface intuitiva facilitando a navegação do usuário direcionando conforme seus objetivos.

Podendo assegurar que realize processos automáticos conforme os dados informados pelo usuário ao fazer o cadastro de paciente.

Concluindo que supere as necessidades da clínica, de acordo com os dados recolhidos no levantamento de requisitos.

REFERENCIAS

DEVMEDIA, E. Conheça o Apache Tomcat. **devmedia.com.br**. Disponível em: <<http://www.devmedia.com.br/conheca-o-apache-tomcat/4546>>. Acesso em: 22 Outubro 2013.

FOX, D. O NetBeans Fonte História Aberta. **developer.com**, 2001. Disponível em: <<http://www.developer.com/java/ent/article.php/631131/The-NetBeans-Open-Source-Story.htm>>. Acesso em: 23 Outubro 2013.

MEDEIROS, H. Introdução ao EclipseLink. **devmedia.com.br**. Disponível em: <<http://www.devmedia.com.br/introducao-ao-eclipselink/29131>>. Acesso em: 22 Outubro 2013.

MEDEIROS, H. Introduzindo o servidor de aplicação Apache Tomcat. **devmedia.com.br**. Disponível em: <<http://www.devmedia.com.br/introduzindo-o-servidor-de-aplicacao-apache-tomcat/27939>>. Acesso em: 22 Outubro 2013.

MENDES, D. R. **Programação Java em Ambiente Distribuído**. Primeira Edição. ed. São Paulo: Novatec Editora, v. 1, 2011.

Olivatto, R.A. A inovação da tecnologia **portaleducacao.com.br**. Disponível em: <<http://www.portaleducacao.com.br/Artigo/Imprimir/15887>>. Acesso em: 17 Setembro 2013.

PACIEVITCH, Y. História do Java. **infoescola.com**. Disponível em: <<http://www.infoescola.com/informatica/historia-do-java>>. Acesso em: 09 Julho 2013.

PALMEIRA, T. V. V. java historia e principais conceitos. **devmedia.com.br**. Disponível em: <<http://www.devmedia.com.br/java-historia-e-principais-conceitos/25178>>. Acesso em: 22 Outubro 2013.

PISA, P. O que é e como usar o MySQL? **techtudo.com.br**, 2012. Disponível em: <<http://www.techtudo.com.br/artigos/noticia/2012/04/o-que-e-e-como-usar-o-mysql.html>>. Acesso em: 23 Outubro 2013.

REDAÇÃO. Conheça um pouco sobre o MySQL. **oficinadanet.com.br**, 2007.
Disponível em:

<http://www.oficinadanet.com.br/artigo/390/conheca_um_pouco_sobre_o_mysql>.

Acesso em: 23 Outubro 2013.

SILVA, M. S. **jQuery - A Biblioteca do Programador JavaScript**. 1 Edição. ed. São Paulo: NovaTec Editora, 2008.