



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

**MARCELO FRANCO**

**SISTEMA GENÉRICO DE AGENDAMENTO**

Assis  
2013

**MARCELO FRANCO**

## **SISTEMA GENÉRICO DE AGENDAMENTO**

Trabalho de Conclusão de Curso apresentado ao Instituto Municipal de Ensino Superior de Assis (IMESA) como requisito de conclusão do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistema,

Orientador: Prof. Esp. Célio Desiró

Assis  
2013

## FICHA CATALOGRÁFICA

FRANCO, Marcelo

Sistema Genérico de Agendamento / Marcelo Franco - 2013.

45 p.

Orientador: Prof. Esp. Célio Desiró

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

1.C#, 2 Visual Studio 2012, 3 Data Entity Framework 5.0, 4 Asp.Net, 5 Bootstrap

CDD: 001.61  
Biblioteca da FEMA

**MARCELO FRANCO**

## **SISTEMA GENÉRICO DE AGENDAMENTO**

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: Prof. Esp. Célio Desiró**

**Examinador: Prof. Me. Douglas Sanches da Cunha**

Assis  
2013

## ***Dedicatória***

Dedico este trabalho primeiramente a Deus, aos meus pais, meus avós que me criaram com muito carinho, a minha namorada Thais, ao meu grande amigo e chefe Fernando Rosa, aos meus amigos, que me acompanharam nesta caminhada e á todos os meus professores, que me ensinaram e incentivaram tanto durante a graduação.

## **AGRADECIMENTOS**

A todos que me ajudaram e sempre e me deram apoio quando eu mais precisei, mas antes de citar algum nome vou agradecer a Deus que me ajudou, me deu força e inteligência para superar minhas dificuldades. A primeira pessoa que me incentivou e me fez prosseguir os estudos, diante de todas as dificuldades e perseverou para o meu sucesso, me apoiando para eu não desistir dos meus sonhos, qualquer que seja, à minha namorada Thais Salatini; ao Nelson Luiz dos Santos que me ajudou no pré-projeto; ao professor Célio Desiró, meu orientador que sempre me deu todo apoio necessário para a realização deste trabalho; ao professor Almir Rogério Camolesi que me ajudou muito na programação; ao meu grande amigo Aduino Francisco Leite Neto, que mesmo distante, me auxiliou e me deu aulas pelo Skype e aos meus amigos Érion Barasuol e Frederico, que me incentivaram a programar. Obrigado a todos.

## RESUMO

Este trabalho defende a ideia de que a interface gráfica é um dos fatores determinantes para o desenvolvimento de um software, senão o mais importante deles, porque afinal, para o usuário a Interface gráfica é o sistema. O sistema de gerenciamento de agendamento será desenvolvido com um framework chamado Bootstrap, ele facilitará no desenvolvimento do sistema e suas interfaces ficarão mais profissionais e usuais. O uso de sistemas computacionais aumentou consideravelmente, junto com esse crescimento a interatividade e a dinâmica das aplicações cresceram também, o apelo visual sempre foi fator determinante para venda e usabilidade de produtos. Com o desenvolvimento de software não é diferente, seja para pequenas ou grandes empresas ou até mesmo para desenvolvedores freelancer, como desenvolver uma boa interface gráfica, propor o desenvolvimento software como produto e um produto de qualidade e proporcionar através de interfaces gráficas bem estruturadas um feedback maior com o usuário de sistemas. O visual do sistema é um requisito básico para conquistar um bom lugar no mercado de desenvolvimento de softwares.

**Palavras-chave:** C#, Visual Studio 2012, Data Entity Framework 5.0, Asp.Net, Bootstrap

## ABSTRACT

*This work supports the idea that the GUI is one of the determining factors for the development of software, but the most important one, because after all, to the user the graphical interface is the system. The management system will be developed with a scheduling framework called Bootstrap, it will facilitate the development of the system and its interfaces will become more professional and usual. The use of computer systems has increased considerably, along with this growth interactivity and dynamic applications will also grow, visual appeal has always been a determinant for sale and usability of products. With the development of software is no different. Whether for small or large companies or even freelance developers to develop a good graphical interface, proposing development software product and as a quality product and provide graphical interfaces through well-structured greater feedback to the user system. It is a basic requirement to get a good place in the market of software development.*

**Keywords:** *C#, Visual Studio 2012, Data Entity Framework 5.0, Asp. Net, Bootstrap*



## LISTA DE ILUSTRAÇÕES

Figura 1: Exemplo de Classe do Visual Studio.....	16
Figura 2: Exemplo de Estrutura do <i>Bootstrap</i> .....	18
Figura 3: Exemplo de <i>Menu Bootstrap</i> .....	19
Figura 4: Código da figura 3.....	19
Figura 5: Modelo Entity Data Model.....	21
Figura 6: Diagrama de Caso de Uso Geral.....	25
Figura 7: Ilustra o caso de Uso Cadastrar Cliente.....	26
Figure 8: Ilustra Diagrama de atividade Cadastrar Cliente.....	27
Figure 8: Ilustra Diagrama de atividade Cadastrar Cliente.....	28
Figura 9: Ilustra o Caso de uso Cadastrar Profissional.....	30
Figura 10: Ilustra o caso de uso Agendar.....	31
Figura 11: Ilustra o Diagrama de Atividade Agendar Cliente.....	32
Figura 12: Caso de uso - Visualizar Agenda.....	33
Figura 13: Diagrama de classe.....	34
Figura 14: Diagrama de classe Cliente.....	35
Figura 85: Diagrama de sequencia Cadastrar Cliente.....	36
Figura 96: Diagrama de Sequencia Agendar Cliente .....	37
Figure 17: Diagrama de Entidade e Relacionamento.....	38
Figure 18: Tela de Autenticação dos usuários.....	41
Figure 19: Tela Principal do sistema.....	42
Figure 20: Tela de cadastro de cliente.....	43
Figure 21: Agendamento de Cliente.....	44

## LISTA DE TABELA

Tabela 1: Tabela de requisitos Funcionais.....	25
Tabela 2: Cronograma de atividades.....	39

## LISTA DE ABREVIATURAS E SIGLAS

ASP - *Active Server Pages*

C# - *Csharp*

CRL - *Common Language Runtime*

EDM - *Entity Data Model*

HTML - *Hyper Text Markup Language*

MSIL - *Microsoft Intermediate Language*

OMG - *Object Management Group*

PDC - *Professional Developers Conference*

SMC - *Simple Managed C*

SQL - *Structured Query Language* ou *Linguagem de Consulta Estruturada*

UML - *Unified Modeling Language*

VB.NET - *Visual Basic .NET*

WEB - *Rede Mundial de Computadores*

XML - *Extensible Markup Language*

## SUMÁRIO

1	INTRODUÇÃO	13
1.2	OBJETIVOS	13
1.3	JUSTIFICATIVAS	14
1.4	MOTIVAÇÃO	14
1.5	ESTRUTURA DO TRABALHO	14
2	TECNOLOGIAS DE ANÁLISE E DESENVOLVIMENTO	15
2.1	PLATAFORMA <i>MICROSOFT .NET</i>	15
2.2	C SHARP	15
2.3	ASP.NET	17
2.4	BOOTSTRAP	17
2.4.2	<i>Estrutura do Bootstrap</i>	18
2.4.3	<i>Navegação básica</i>	19
2.5	ENTITY FRAMEWORK	20
3	TECNOLOGIAS E FERRAMENTAS PARA A PREPARAÇÃO DA ANÁLISE	22
3.1	LINGUAGEM DE MODELAGEM UML	22
3.1.1	<i>Astah UML</i>	22
4	LEVANTAMENTO DOS REQUISITOS	23
4.1	FORMA ADOTADA	23
5	ANÁLISE DOS REQUISITOS	24
5.1	REQUISITOS FUNCIONAIS	24
5.2	REQUISITOS NÃO FUNCIONAIS	24
5.2.1	<i>Requisito de segurança</i>	24
5.2.2	<i>Requisitos desempenho</i>	25
5.2.3	<i>Requisitos de facilidade de uso</i>	25
5.2.4	<i>Diagrama de Caso de Uso</i>	25
5.2.5	<i>Cadastrar Cliente</i>	26
5.2.6	<i>Casos de atividade</i>	27
5.2.7	<i>Cadastrar Profissional</i>	28
5.2.9	<i>Agendar Cliente</i>	30
5.2.10	<i>Diagrama de atividade agendar cliente.</i>	32
5.2.11	<i>Visualizar Agenda</i>	33
6	DIAGRAMA DE CLASSE	34
6.1	DIAGRAMA DE CLASSE CLIENTE	35
7	DIAGRAMA DE SEQUENCIA CADASTRAR CLIENTE	36
7.1.1	<i>Agendar Cliente</i>	37
8	DIAGRAMA DE ENTIDADE E RELACIONAMENTO	38
9	CRONOGRAMA	39
	CONCLUSÃO	40
	REFERÊNCIAS	41
	ANEXOS	42

## 1 INTRODUÇÃO

O crescimento de aplicações *WEB* está fortemente ligado à necessidade de agilidade e facilidade, proporcionada aos usuários nas aplicações e facilitando as atualizações: atualizações que serão feitas direto no servidor em pouco tempo, agilizando o acesso do usuário e deixando o ambiente de trabalho mais rápido. Esse tipo de sistema tem se destacado, pois são aplicações que podem se tornar independentes do ambiente de implantação. Neste caso, não precisa estar instalado na máquina do usuário, mas hospedado em um servidor. O usuário poderá ter acesso ao sistema de qualquer lugar, facilitando o seu trabalho. Para que isto funcione o usuário deverá estar conectado à *internet*.

O sistema de agendamento a ser implementado é genérico e será desenvolvido para atender a qualquer tipo de empresa, proporcionando facilidade e rapidez em seus atendimentos via *web*.

### 1.2 Objetivos

O objetivo do sistema é facilitar o trabalho das empresas que necessitam de acesso ágil via *web*.

Bem aplicado, o sistema agilizará o processo de gerenciamento de agendamento e facilitará o atendimento, proporcionando mais segurança para as empresas e seus clientes.

Com um acesso via *web* o profissional poderá acessar sua agenda de trabalho diariamente, o que facilitará o tipo de atendimento que realizará aos seus clientes e/ou deixando de ocasionar a falta de equipamentos para o profissional no atendimento.

O principal foco do sistema proposto é a automatização das etapas presentes na rotina de alguns consultórios médicos e odontológicos. Visando a organização dos processos, bem como, tornando-os mais ágeis, reduzindo despesas e inibindo a ocorrência de falhas humanas. Assim, contribuindo de maneira direta para a lucratividade da organização.

### **1.3 Justificativas**

O desenvolvimento *web* vem crescendo muito, um sistema via *web* irá facilitar e agilizar o sistema de agendamento de empresas que utilizam esse tipo de serviço em *desktop*. Com o desenvolvimento de sistema *web* alguns problemas como de atualizações, não serão mais parte dos problemas de uma empresa que utiliza desse tipo de serviços diariamente, pois não precisarão parar o sistema atual para a sua atualização.

### **1.4 Motivação**

Nesta linha de raciocínio, nota-se que o sistema a ser desenvolvido nos consultórios e empresas irá melhorar o atendimento aos seus pacientes/clientes, tendo assim mais facilidade nos agendamentos de consultas.

### **1.5 Estrutura do Trabalho**

Para melhor organização, este trabalho foi desenvolvido em capítulos.

O capítulo 1 refere-se a introdução. Já no capítulo 2 serão apresentadas as tecnologias de análises e desenvolvimento do sistema, assim como uma breve descrição de cada uma delas. No capítulo 3, será levantado os requisitos para o desenvolvimento do sistema. Em sequencia terá o capítulo 4 que serão as análises dos requisitos necessários para o funcionamento do sistema. O capítulo 5 mostrará o diagrama de entidade e relacionamento e por fim, o capítulo 6 mostrará a tabela do cronograma das atividades.

## 2 TECNOLOGIAS DE ANÁLISE E DESENVOLVIMENTO

Nesse capítulo serão apresentadas as ferramentas para o desenvolvimento do sistema e suas plataformas.

### 2.1 Plataforma *Microsoft .NET*

Segundo Alvarez (2004), a *Microsoft .NET* ou *.NET Framework* como é conhecido, é uma iniciativa da empresa Microsoft, que visa uma plataforma única para desenvolvimento e execução de sistemas e aplicações. Todo e qualquer código gerado para .NET, pode ser executado em qualquer dispositivo que possua um *framework* de tal plataforma.

Essa plataforma é executada sobre uma *Common Language Runtime* (CLR), que é capaz de executar atualmente mais de 20 diferentes linguagens, interagindo entre si como se fossem uma única linguagem. A plataforma .NET baseia-se em um dos princípios utilizados na tecnologia Java, os programas desenvolvidos para ela são duplo-compilados (compilados duas vezes), uma na distribuição (gerando um código que é conhecido como "*bytecodes*") e outra na execução. Um programa é escrito em qualquer das mais de vinte linguagens de programação disponíveis para a plataforma, o código fonte gerado pelo programador é então compilado pela linguagem escolhida gerando um código intermediário em uma linguagem chamada *Microsoft Intermediate Language* (MSIL).

### 2.2 C Sharp

De acordo com Alvarez (2004), o *C Sharp* (C#) é uma linguagem de programação orientada a objetos, fortemente tipada desenvolvida pela Microsoft como parte da plataforma .NET. A sua sintaxe orientada a objetos foi baseada no C++ mas inclui muitas influências de outras linguagens de programação, como *Object Pascal* e Java. Durante o desenvolvimento da plataforma .NET, as bibliotecas foram escritas originalmente em uma linguagem chamada *Simple Managed C* (SMC), que tinha um

compilador próprio. Mas, em janeiro de 1999, uma equipe de desenvolvimento foi formada por Anders Hejlsberg, que fora escolhido pela Microsoft para desenvolver a linguagem. Dá-se início à criação da linguagem chamada *Cool*. Um pouco mais tarde, em 2000, o projeto .NET era apresentado ao público na *Professional Developers Conference* (PDC), e a linguagem *Cool* fora renomeada e apresentada como C#.

A criação da linguagem, embora tenha sido feita por vários programadores, é atribuída principalmente a Anders, hoje um *Distinguished Engineer* na Microsoft. Ele fora o arquiteto de alguns compiladores da *Borland*, e entre suas criações mais conhecidas estão o *Turbo Pascal* e o *Delphi*. Apesar de o C# ser tido como similar a Java, existe uma série de diferenças importantes, mas a maioria é implementada de diferentes formas em ambas as linguagens.

Exemplo de uma Classe do C# escrita no Visual Studio:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;

namespace SistemaAgendamentoWeb
{
    public class ClasseTeste
    {

    }
}
```

Figura 10: Exemplo de Classe do Visual Studio



## 2.3 Asp.NET

*ASP.NET* é a plataforma da Microsoft para o desenvolvimento de aplicações *Web* e é o sucessor da tecnologia ASP. É um componente do IIS que permite através de uma linguagem de programação integrada ao *.NET Framework* criar páginas dinâmicas.

O *ASP.NET* é baseado no *Framework .NET* herdando todas as suas características, por isso, como qualquer aplicação *.NET*, as aplicações para essa plataforma podem ser escritas em várias linguagens, como C# e *Visual Basic .NET* (VB.NET).

## 2.4 Bootstrap

Desde os guias de *design*, até os padrões de interação, se tem a eterna necessidade de definir os padrões e a consistência em uma aplicação *web* de modo a reutilizar o máximo possível os elementos de *design*. Se não bastasse isto, o *Twitter*, com o *Bootstrap* foi além e implementou um *framework* onde há todos os elementos de um *website*, seguindo todo o estilo padrão do estilo do *Twitter* e *Open Source*. Eles o chamaram de *Bootstrap* e foi criado por Mark Otto e Jacob Thornton. Quais as vantagens de ter o *Bootstrap* no início do projeto ou até mesmo usar o *Bootstrap* do *Twitter*?

Uma das principais vantagens de utilizar o *Bootstrap* é que ele padroniza todos os elementos de design da tela, ou seja, ele não modifica o padrão do visual do navegador como o Chrome, Explorer ou Mozilla e sim adequa a qualquer tipo de navegador, até mesmo no celular móvel.

Usar o próprio *Bootstrap* do *Twitter* para prototipar é também algo extremamente útil, pois ele é bastante simples, limpo e com todos os recursos de CSS3.

No entanto é necessário ter no mínimo alguns itens: Elementos de grid, Formulários, Tipografia, Cabeçalhos, Listas, Tabelas, Botões.

### 2.4.1 Estrutura do Bootstrap

A figura abaixo representa a estrutura dos *templates* utilizados no *Hyper Text Markup Language* (HTML Básico) e incluindo referencias de CSS e *Java Script* para iniciar uma interface do sistema com o *Bootstrap*.

```
1. <html>
2.   <head>
3.     <title>Bootstrap Template</title>
4.     <!--Bootstrap referencias -->
5.     <link href="css/bootstrap.min.css"
6.         rel="stylesheet">
7.   </head>
8.   <body>
9.     <!--Bootstrap titulo -->
10.    <h1> Hello, world!</h1>
11.    <script
12.        src="js/bootstrap.min.js"></script>
13.  </body>
14. </html>
```

Figura 11: Exemplo de Estrutura do *Bootstrap*

## 2.4.2 Navegação básica

Para começar, as barras de navegação são estáticas (não fixas no topo) e incluem suporte para um nome de projeto para navegação básica. No *site* <http://twitter.github.io/bootstrap/> tem todos os comando e códigos necessários para a construção de *sites* e *web server*.

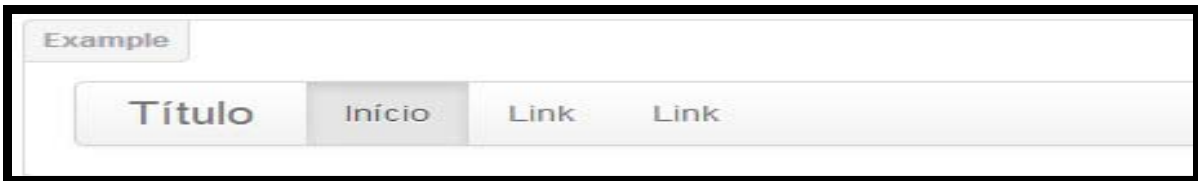


Figura 12: Exemplo de *Menu Bootstrap*

A figura abaixo mostra o código fonte da imagem acima.

```

1. <div class="navbar">
2.   <div class="navbar-inner">
3.     <a class="brand" href="#">Título</a>
4.     <ul class="nav">
5.       <li class="active"><a
6.         href="#">Início</a></li>
7.       <li><a href="#">Link </a></li>
8.       <li><a href="#">Link </a></li>
9.     </ul>
10.  </div>
11. </div>

```

Figura 13: Código da figura 3

## 2.5 Entity Framework

O *Entity Data Model* (EDM) é um modelo de dados Entidade-Relacionamento cujo conceito central são as entidades e os relacionamentos, onde Entidades são instâncias de Tipos de entidades como Cliente, Produto, Categoria, as quais são estruturas de registros com uma chave.

Uma Chave de Entidade é formada a partir de um subconjunto de propriedades do Tipo da entidade.

A chave (*clienteID*, *ProdutoID*, etc) é um conceito fundamental para identificar de forma única, atualizar instâncias e permitir que as instâncias de entidades participem nos relacionamentos.

Além do *Entity Data Model*, *Entity Framework* inclui serviços que incrementam o poder do modelo incluindo o provedor gerenciado *EntityClient ObjectServices* e *LINQ to Entities*.

O *Object Services* é um componente do *Entity Framework* que permite que você realize consultas, inclusões, atualizações e exclusões de dados, expressos como um objeto CLR fortemente tipados que são instâncias de tipos de entidades. O mesmo dá suporte a consultas LINQ e *Entity SQL* contra tipos que são definidos em um EDM.

O *Object Services* também materializam os dados retornados como objetos e propagam as alterações dos objetos de volta a fonte de dados, além de fornecer facilidades para tratar alterações, vinculação de controles a objetos e efetuar o tratamento da concorrência.

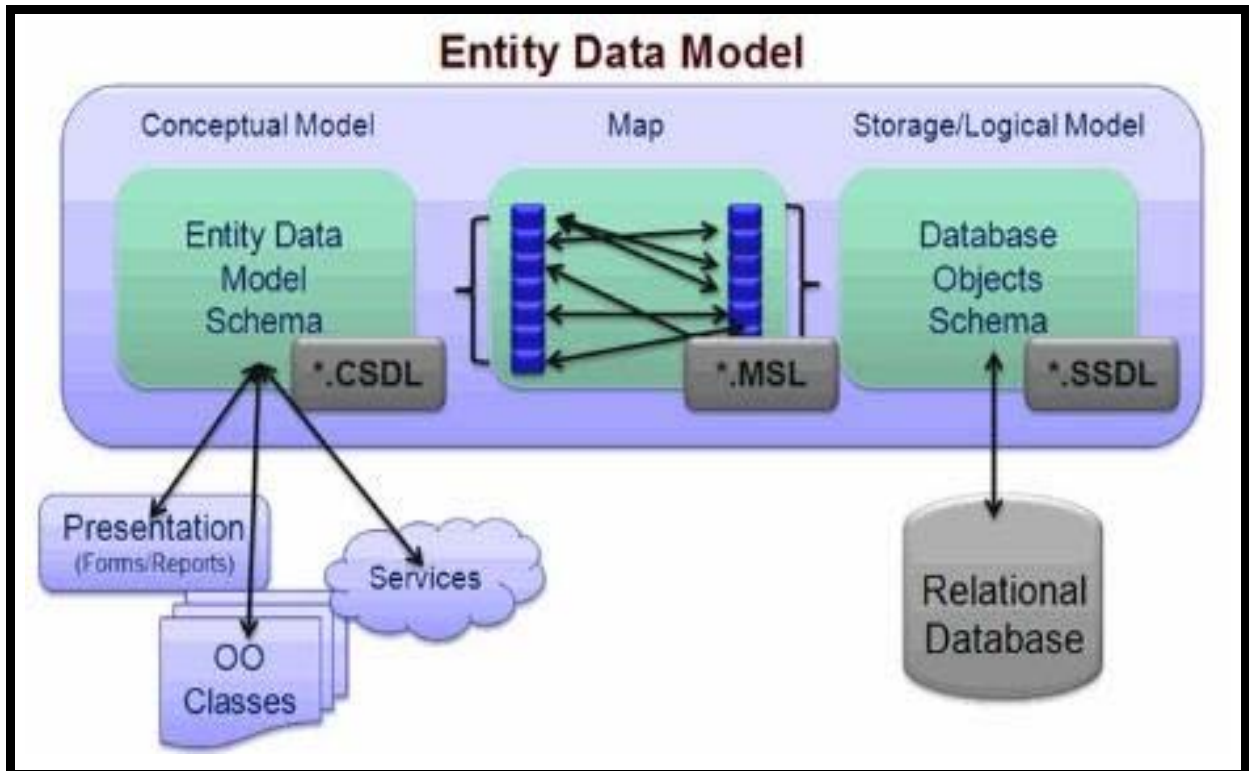


Figura 14: Modelo Entity Data Model

- Arquivo de definição de Schema conceitual - (.CSDL) - define o modelo conceitual;
- Arquivo de definição Schema de armazenamento - (.SSDL) - Define o modelo de armazenamento que também é chamado de modelo lógico;
- Arquivo de especificação de mapeamento - (.MSL) - Define o mapeamento entre os modelos conceitual e de armazenamento;

O *Entity Framework* utiliza estes modelos baseados em *Extensible Markup Language* (XML) e arquivos de mapeamento para transformar operações de criar, ler, atualizar, e deletar contra entidades e relacionamentos no modelo conceitual para operações equivalentes na fonte de dados.

### **3 TECNOLOGIAS E FERRAMENTAS PARA A PREPARAÇÃO DA ANÁLISE**

#### **3.1 Linguagem de modelagem UML**

A *Unified Modeling Language* – Linguagem de Modelagem Unificada (UML) é uma linguagem que foi desenvolvida pela *Object Management Group* (OMG), em 1997, para criação da modelagem da análise de um projeto, ajudando a tornar o sistema melhor documentação e padronização. A UML foi adotada por permitir definições de sistema orientados a projeto bem definidos, se tornando uma notação padrão na diagramação da modelagem de projetos.

##### **3.1.1 Astah UML**

Para demonstrar a visualização dos diagramas foi utilizada a ferramenta Astah UML, que trata-se de um editor que facilita a representação das tabelas, diagramas e seus relacionamentos. O Astah é utilizado no desenvolvimento de processos onde se necessita uma representação gráfica das partes envolvidas.

## 4 LEVANTAMENTO DOS REQUISITOS

Neste capítulo, será levantado os requisitos para o desenvolvimento de um sistema. Para efetuar um sistema bem feito será utilizado requisitos do UML, que ajudará num desenvolvimento e organizará meu sistema em etapas que facilitarão na modelagem.

### 4.1 Forma Adotada

Os requisitos foram levantados de um estudo, a fim de suprir as necessidades de empresas. Durante o estudo foram levantados alguns requisitos, como:

Cadastro de funcionários, clientes, Profissional, Administrador, tipo de agendamento.

Realizar agendamento.

Relatórios do histórico de clientes e agendamentos.

Cadastros: A finalidade dos cadastros é ter informações seguras, de modo para que não perca as mesmas.

Agendamento: Irá ajudar o profissional a controlar e organizar todas os atendimentos que o mesmo terá no dia-a-dia.

Relatórios: Os relatórios tem finalidade para ter um controle em tudo que foi feito tanto no dia como no mês com todos serviços realizados.

## 5 ANÁLISE DOS REQUISITOS

Após as discussões com o orientador Celio Desiró foram identificados os requisitos necessários para o funcionamento do sistema e serão apresentados abaixo.

### 5.1 Requisitos Funcionais

Requisitos funcionais são ações do usuário com o sistema, requisitos estes que deverão ser atendidos pelo sistema.

Número	Descrição	Caso de uso
01	Cadastrar (Cliente, Usuário, Profissional)	Cadastrar profissional, Cadastrar Usuário, Cadastrar Cliente.
02	Agendar Cliente	Agendar
03	Visualizar Cliente	Visualizar Agenda
04	Registrar Atendimento	Registrar Atendimento
05	Registrar tipo Atendimento	Registrar tipo Atendimento
06	Visualizar Agenda	Visualizar Agenda
07	Visualizar Agenda	Visualizar Agenda

**Tabela 2: Tabela de requisitos Funcionais**

### 5.2 Requisitos não Funcionais

Requisitos não funcionais são aquele que não são feito pelo usuário mais sim pelo sistema.

#### 5.2.1 Requisito de segurança

Todo usuário do sistema deverá efetuar o *login* no sistema, deverá fornecer um nome de *login* e uma senha para ter acesso no sistema, cada usuário terá seu acesso diferenciado e suas restrições em algumas ações no sistema.



### 5.2.2 Requisitos desempenho

O sistema deverá responder no tempo necessário em cada ação do usuário.

### 5.2.3 Requisitos de facilidade de uso

O sistema terá um *menu* principal que facilitará nas ações do usuário no sistema.

### 5.2.4 Diagrama de Caso de Uso

A Figura 6 ilustra o caso de uso que pode ser representado por uma elipse contendo, internamente, o nome do caso de uso.

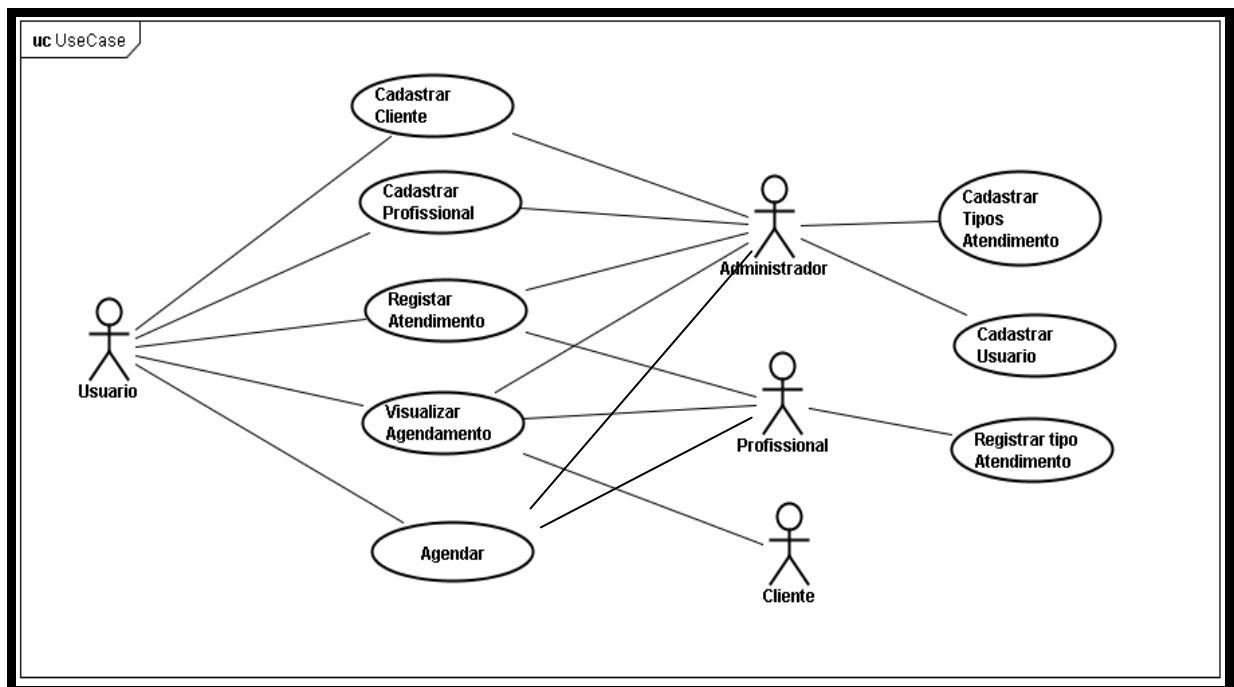


Figura 15: Diagrama de Caso de Uso Geral

## 5.2.5 Cadastrar Cliente

A figura abaixo apresenta o Agendar Cliente, onde o usuário e profissional poderá estar preenchendo os campos e agendando o cliente.

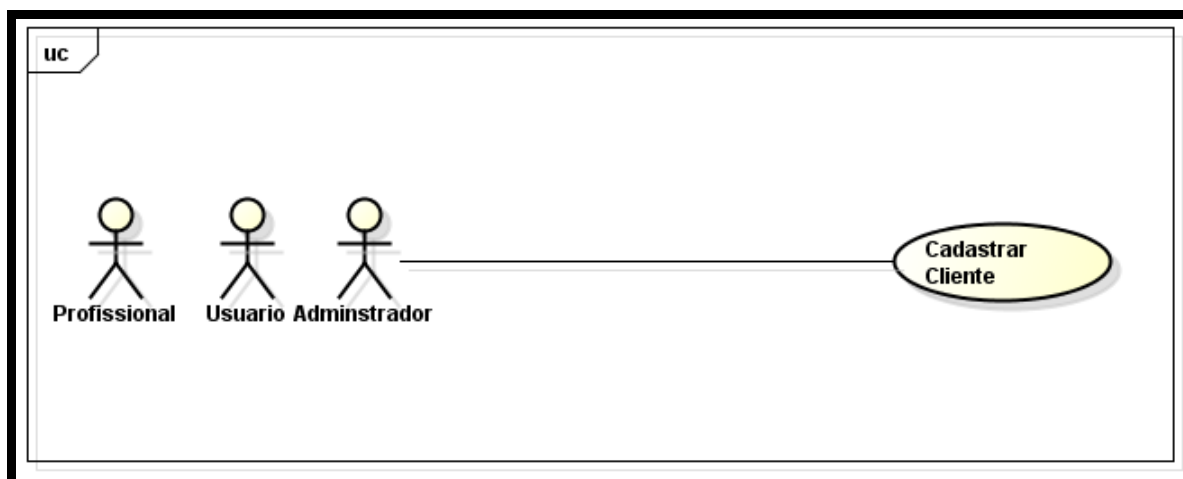


Figura 16: Ilustra o caso de Uso Cadastrar Cliente

### 1. Finalidade/Objetivo

Permite ao usuário cadastrar todos os Clientes no sistema.

### 2. Ator

Usuário/Administrador/Profissional

### 3. Precondições

O usuário deve ter efetuado *login* como Administrador

### 4. Fluxo principal

a. O usuário seleciona a opção cadastrar clientes no *menu* principal e em seguida escolhe a opção cadastrar Cliente.[A1].[A2]. b. O sistema solicita os dados necessários para o cadastro do Cliente. c. O usuário fornece os dados e confirma a operação.[A3]. d. O sistema verifica se todos foram fornecidos e em seguida mostra uma mensagem de confirmação.

**5. Fluxo Alternativo A1.** O usuário seleciona a opção Excluir cliente.

O sistema solicita o código do cliente a ser excluído.

**A2.** O usuário escolhe a opção Alterar cliente.

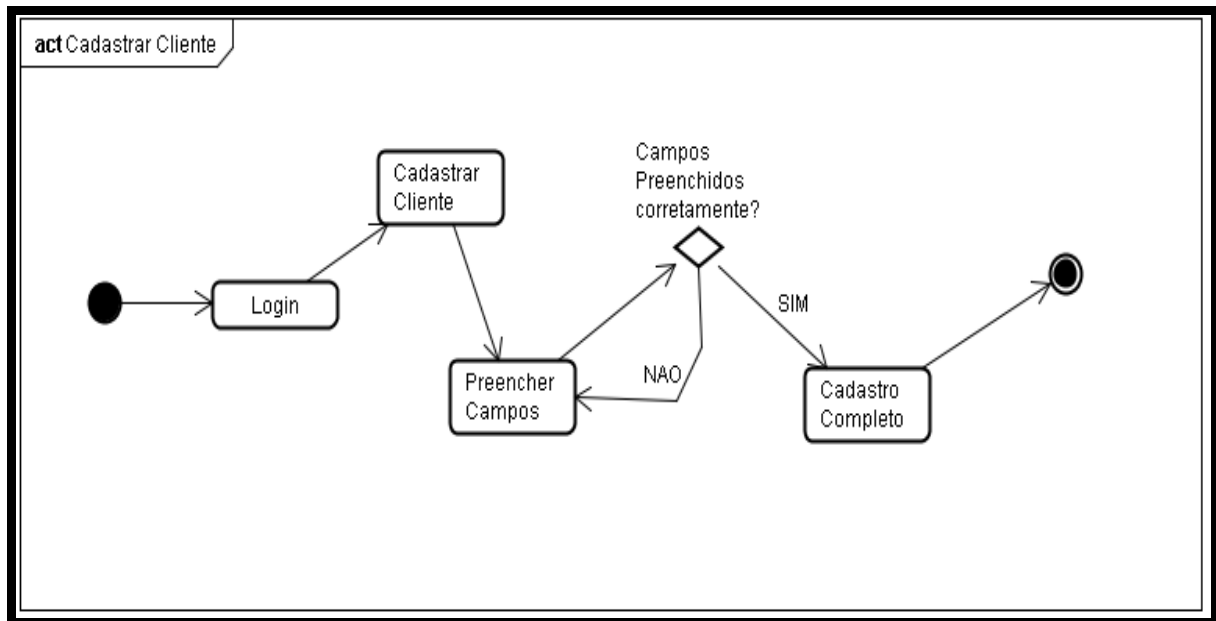
O sistema solicita o código do cliente a ser alterado.

**A3.** O usuário escolhe a opção Cancelar.

O sistema volta ao passo 4.a do Fluxo Principal.

## 5.2.6 Casos de atividade

Diagrama de atividade de Cadastrar Cliente.

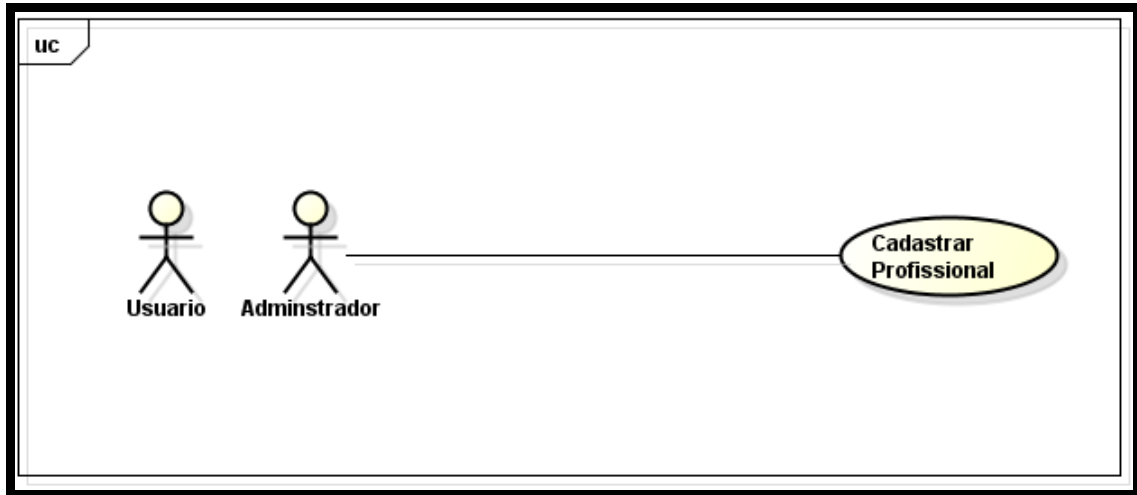


**Figure 8: Ilustra Diagrama de atividade Cadastrar Cliente**

O cliente faz *login* no sistema, clica no botão cadastrar -> cadastrar cliente, preenche os campos e salva o cadastro.

### 5.2.7 Cadastrar Profissional

A figura abaixo apresenta o caso de uso usuário Cadastrar Profissional, onde o usuário poderá cadastrar o profissional preenchendo todos os campos necessários.



**Figura 9: Ilustra o Caso de uso Cadastrar Profissional**

#### 1. Finalidade/Objetivo

Permite ao usuário cadastrar todos os Profissional no sistema.

#### 2. Ator

Usuário/administrador

#### 3. Precondições

O usuário deve ter efetuado *login*.

#### 4. Fluxo principal

a. O usuário seleciona a opção cadastrar Profissional no *menu* principal e em seguida escolhe a opção cadastrar Profissional.[A1].[A2]. b. O sistema solicita os dados necessários para o cadastro do Profissional c. O usuário fornece os dados e confirma a operação.[A3]. d. O sistema verifica se todos foram fornecidos e em seguida mostra uma mensagem de confirmação.

#### 5. Fluxo Alternativo

**A1.** O usuário seleciona a opção Excluir Profissional.

O sistema solicita o código do profissional a ser excluído.

**A2.** O usuário escolhe a opção Alterar Profissional.

O sistema solicita o código do Profissional a ser alterado.

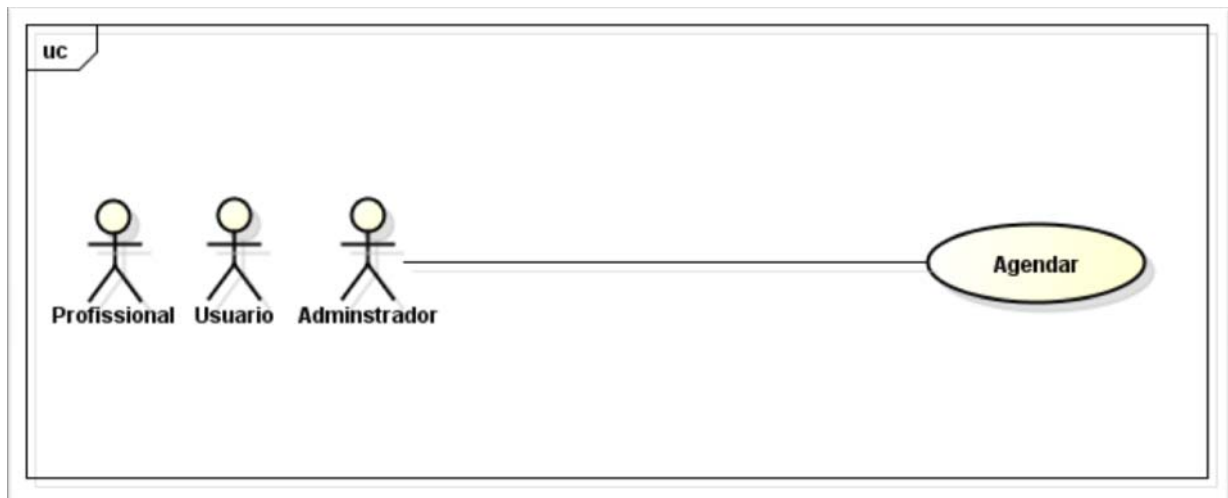
**A3.** O usuário escolhe a opção Cancelar.

O sistema volta ao passo 4.a do Fluxo Principal.

### **6. Casos de Teste**

Verificar se todos os campos foram preenchidos corretamente (campos Numéricos, data e etc).

### 5.2.8 Agendar Cliente



**Figura 10: Ilustra o caso de uso Agendar**

#### 1. Finalidade/Objetivo

Permite ao usuário agendar para qualquer cliente no sistema.

#### 2. Ator

Usuário/profissional/administrador

#### 3. Precondições

O usuário deve ter efetuado *login*.

#### 4. Fluxo principal

- a. O usuário seleciona a opção Agendar no *menu* principal e em seguida escolhe a opção cadastrar Agenda.[A1].[A2].
- b. O sistema solicita os dados necessários para o agendamento do cliente
- c. O usuário fornece os dados e confirma a operação.[A3].
- d. O sistema verifica se todos os dados foram fornecidos e em seguida mostra uma mensagem de confirmação.

#### 5. Fluxo Alternativo

**A1.** O usuário seleciona a opção Excluir Agendamento.

O sistema solicita o código do agendamento a ser excluído.

**A2.** O usuário escolhe a opção Alterar Agendamento.

O sistema solicita os dados para a alteração dos agendamento.

**A3.** O usuário escolhe a opção Cancelar.

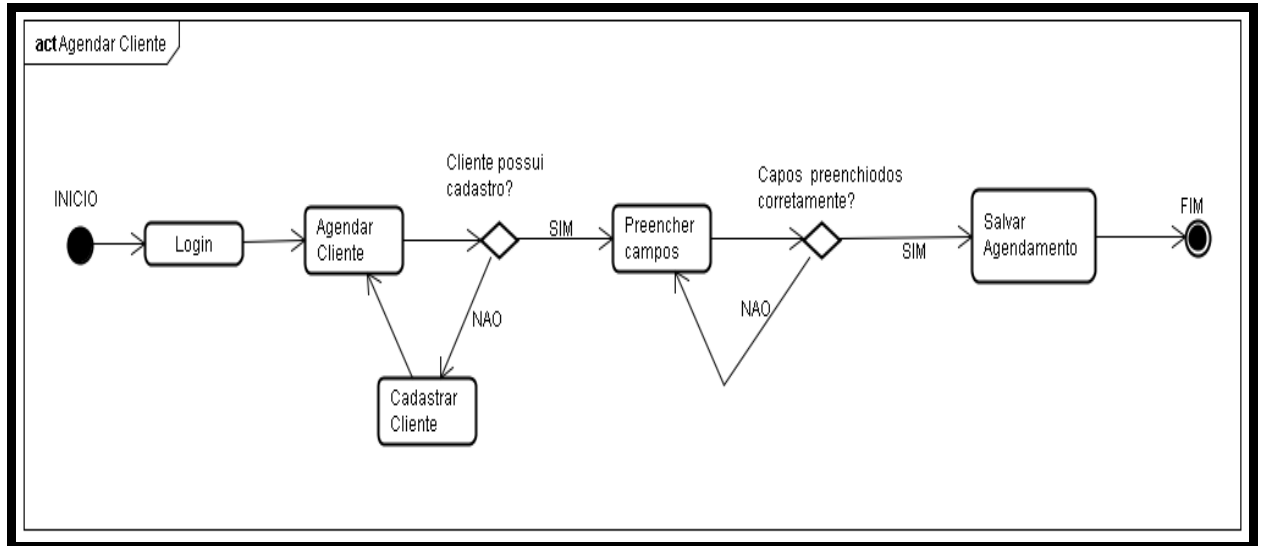
O sistema volta ao passo 4.a do Fluxo Principal.

## **6. Casos de Teste**

Verificar se todos os campos foram preenchidos corretamente (campos Numéricos, data e etc).

### 5.2.9 Diagrama de atividade Agendar Cliente.

A figura abaixo representa o diagrama de atividade Cadastrar Cliente.



**Figura 11: Diagrama de Atividade Agendar Cliente**

O cliente faz *login* no sistema, acessa o ícone no *menu* principal “agendar”, escolhe o cliente a ser agendado, preenche os campos necessários para o agendamento salva e finaliza.



### 5.2.10 Visualizar Agenda

A figura abaixo apresenta o caso de uso Visualizar Agenda, onde o profissional poderá consultar o agendamento do mês.

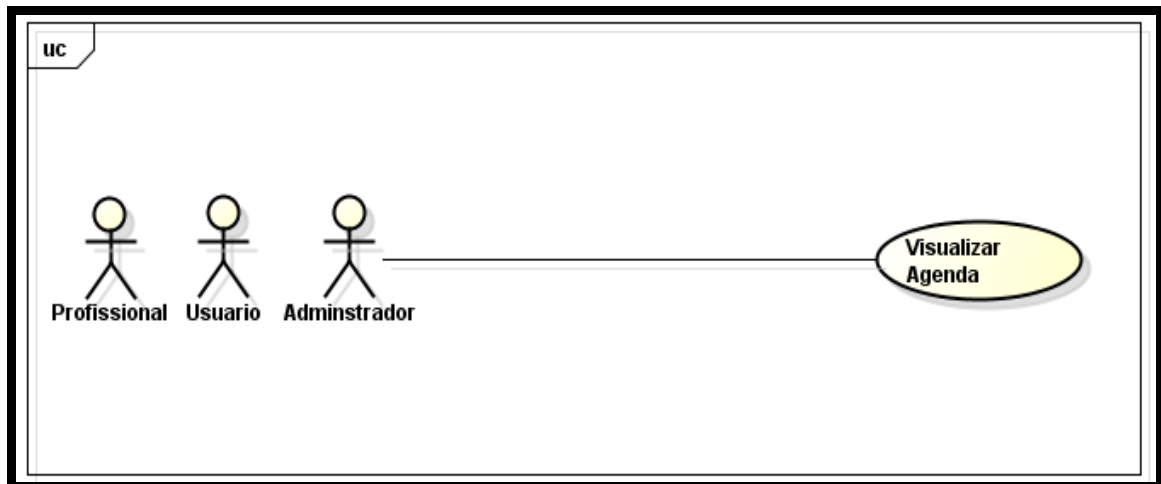


Figura 12: Caso de uso - Visualizar Agenda

#### 1. Finalidade/Objetivo

Permite ao Profissional visualizar todos os clientes agendados no sistema.

#### 2. Ator

Profissional/Usuário/Administrador

#### 3. Precondições

O Profissional deve ter efetuado *login*.

#### 4. Fluxo principal

O profissional seleciona a opção visualizar no *menu* principal e em seguida escolhe a opção Agenda.

#### 5. Fluxo Alternativo

O profissional escolhe a opção Cancelar.

#### 6. Casos de Teste

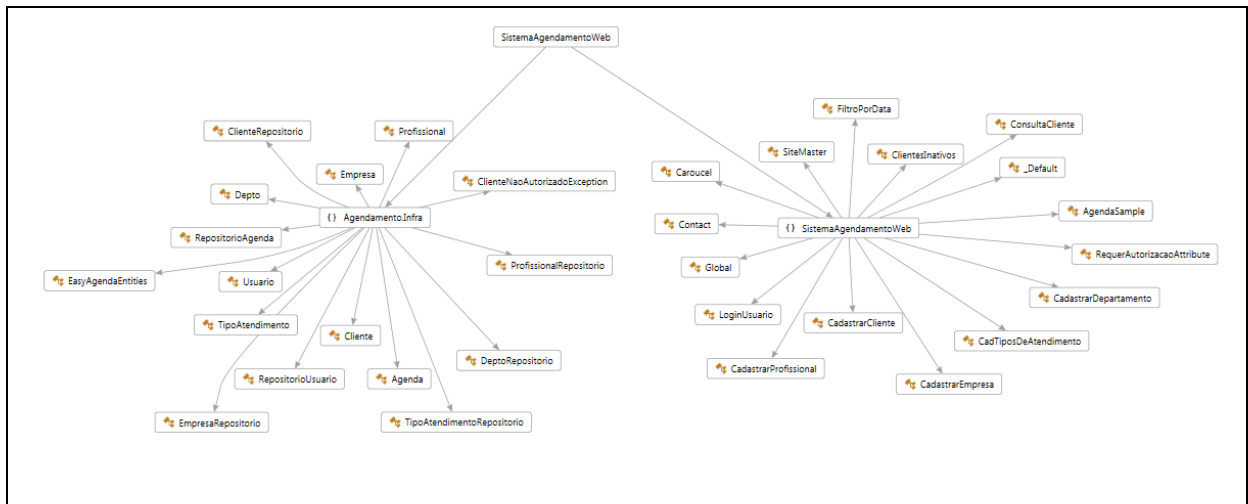
Verificar se todos acessos para as visualizações estão sendo acessados corretamente.

## 6 Diagrama de Classes

A figura abaixo representa todas as classes que utilizadas no sistema.

*AgendamentoInfra* representa todas as regras de negócios, de acesso ao banco e métodos como insert, update e delete.

*SistemaAgendamentoWeb* representa todas as telas paginas que foram construídas para interface.



**Figura 13: Diagrama de classes**



## 7 Diagrama de Sequência Cadastrar Cliente

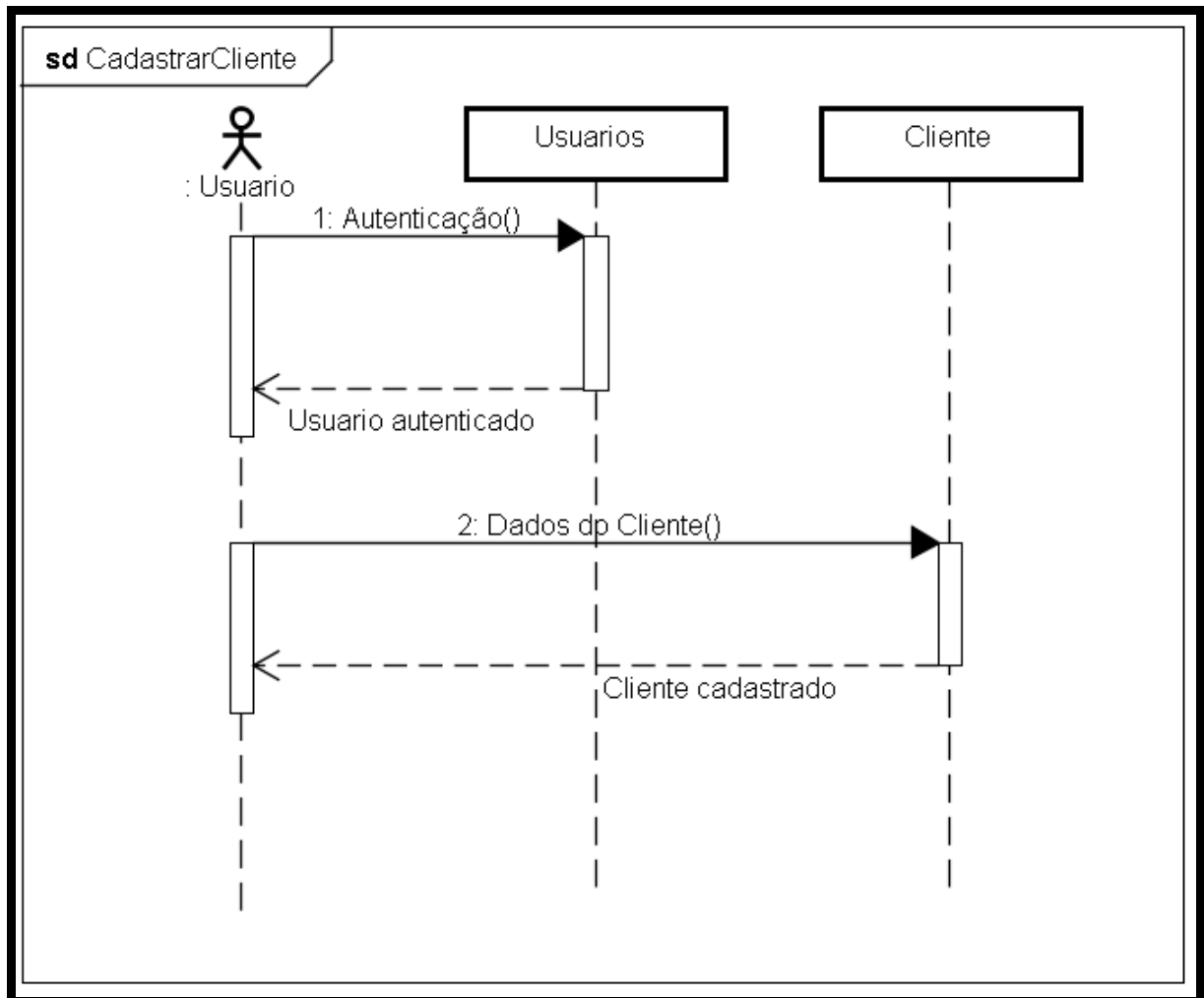


Figura 175: Diagrama de sequência Cadastrar Cliente

### 7.1.1 Agendar Cliente

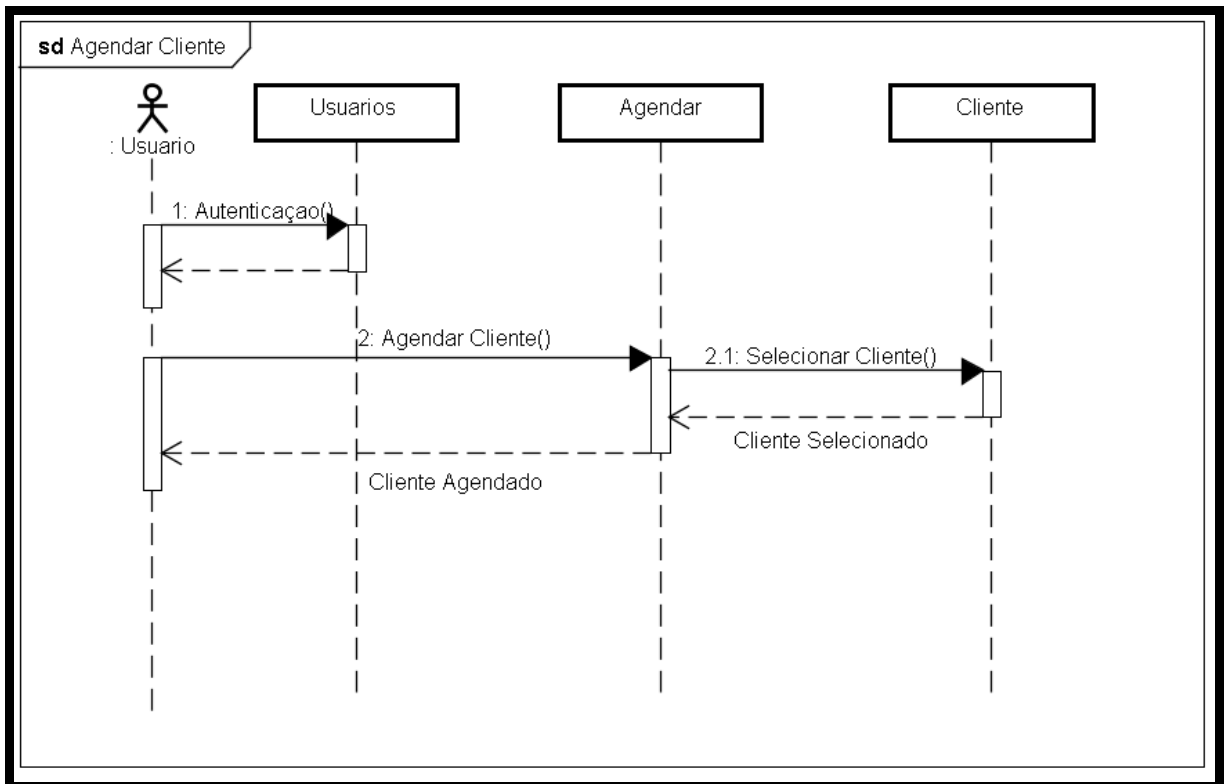


Figura 186: Diagrama de Sequencia Agendar Cliente

## 8 Diagrama de Entidade e Relacionamento

O diagrama é o mapeamento das relações das tabelas do banco de dados, ou seja, o Diagrama de Entidade e Relacionamento é o local onde se armazena todos os dados e informações do sistema.

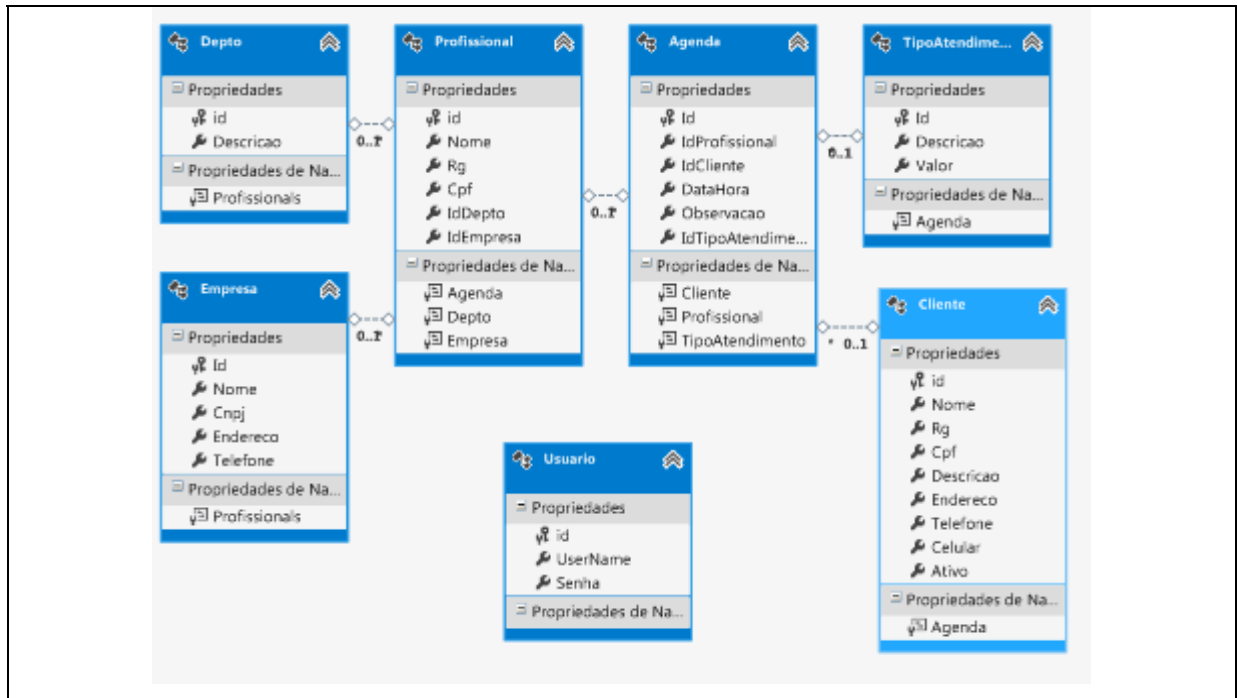


Figure 17: Diagrama de Entidade e Relacionamento

## 9 CRONOGRAMA

Segue abaixo o cronograma das atividades realizadas durante o ano de 2013.

<b>Tarefas</b>	<b>Fev</b>	<b>Mac</b>	<b>Abr</b>	<b>Mai</b>	<b>Jun</b>	<b>Jul</b>	<b>Ago</b>	<b>Set</b>	<b>Out</b>	<b>Nov</b>
Definir Sistema	■									
Levantamento dos requisitos		■								
Análise dos Requisitos		■								
Gerenciamento dos requisitos			■							
Pesquisa do Banco/Framework E modificações			■						■	
Declaração de Objetivos			■							
Diagrama de Casos de Uso				■	■					
Diagrama de Sequência					■					
Diagrama de Atividade					■					
Conclusão					■	■	■			
Bibliografia							■			
Anexos							■			
Desenvolvimento								■	■	
Defesa										■

**Tabela 2: Cronograma de atividades**

## CONCLUSÃO

Neste trabalho foram utilizados os diagramas da UML que foram de extrema importância, pois proporcionaram mais facilidade no desenvolvimento do trabalho com seus relacionamentos e comportamentos dos atributos e métodos de cada classe representada também no banco de dados.

A programação em camadas de projeto facilitou o entendimento de toda infraestrutura do trabalho, padronizou e organizou o código da aplicação e permitiu mais flexibilidade no desenvolvimento.

O contato com a tecnologia do *Bootstrap* proporcionou a aprendizagem e desenvolvimento em aplicações Web, facilitando o desenvolvimento desse projeto e atingindo a satisfação, seja pelos ricos gráficos de sua interface, quanto pela eficiência e facilidade na codificação utilizando o IDE Microsoft Visual Studio 2012.

Com o *Entities Framework* foi mais rápido o desenvolvimento utilizando-se de seus métodos, que proporcionaram mais rapidez à aplicação e o mais importante, o aproveitamento dessa ferramenta tão rica.



## REFERÊNCIAS

ALVAREZ, Miguel Angel. **O que é .NET - A plataforma de Microsoft orientada à criação de Software para Internet**. CriarWeb. Disponível em: <http://www.criarweb.com/artigos/226.php>. Acesso em: 06/06/2013.

BAPTISTA, Luciana Ferreira. **Linguagem SQL – Um guia prático de aprendizagem**. 1 ed. São Paulo. Érica. 2011.

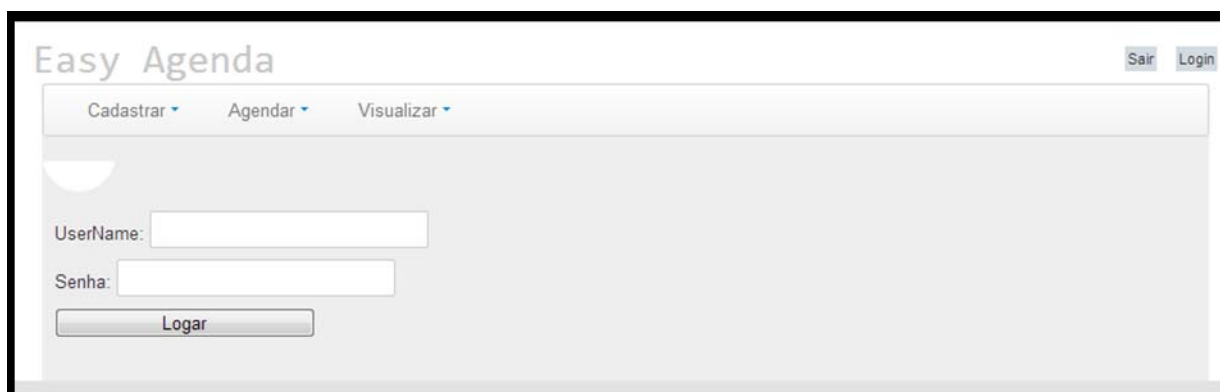
HEUSER, Carlos Alberto. **Projeto de banco de dados**. 6 ed. Porto alegre. Bookman. 2009.

LOTAR, Alfredo. **Como programar em ASP.NET e C#**. 2. ed. São Saulo. Novatec. 2010.

MAGNO, Alexandre. **Vamos começar por Bootstrap**. Webpoint. Disponível em: <http://blog.alexandremagno.net/2011/09/vamos-comecar-por-bootstrap/>. Acesso em: 08/06/2013.

## ANEXO I – Telas Principais do Sistema

Tela de autenticação do usuário, utilizada para realizar o acesso ao sistema.



The screenshot displays the user authentication interface for the 'Easy Agenda' system. At the top left, the title 'Easy Agenda' is visible. In the top right corner, there are two buttons: 'Sair' and 'Login'. Below the title, a navigation bar contains three menu items: 'Cadastrar', 'Agendar', and 'Visualizar', each with a small downward arrow indicating a dropdown menu. The main area of the screen is a light gray color and contains the login form. On the left side of this area, there is a white semi-circular decorative element. The form consists of two input fields: the first is labeled 'UserName:' and the second is labeled 'Senha:'. Below these fields is a single button labeled 'Logar'.

Figure 18: Tela de Autenticação dos usuários

## Tela Inicial do sistema

A figura abaixo apresenta a tela inicial do sistema, após a autenticação do usuário.

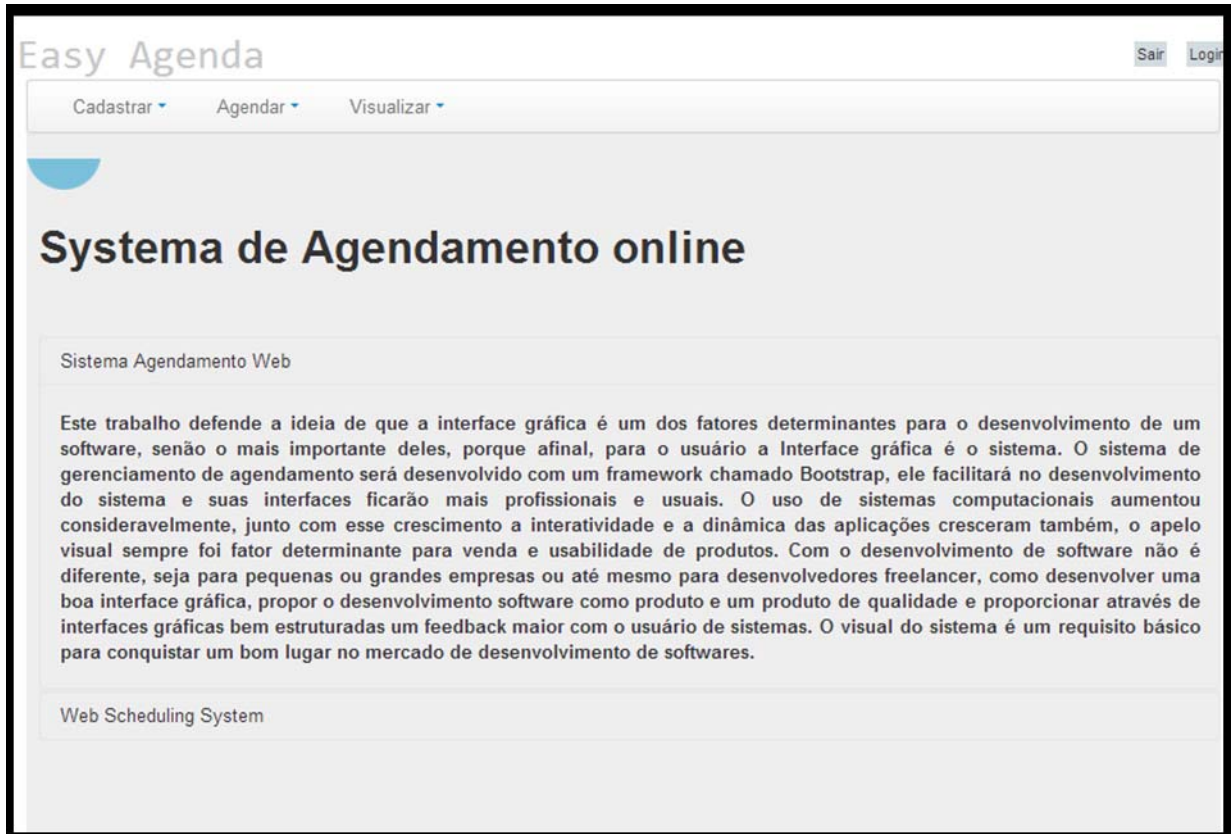


Figure 19: Tela Principal do sistema.

## Cadastrar Cliente

A figura abaixo apresenta a tela de cadastro de clientes, acessada através do menu Cadastrar -> Cliente.

Easy Agenda Sair Login

Cadastrar ▾ Agendar ▾ Visualizar ▾

### Cadastrar Cliente

Nome:

RG:

CPF:

Descrição:

Endereço:

Telefone:

Celular:

[Salvar](#) [Alterar](#) [Salvar Usuario](#)

		id	Nome	Rg	Cpf	Descricao	Endereco	Telefone	Celular	Ativo
Excluir	Selecionar	62	Luiz Salatini	56456456	65456456	Turvo	Bar	321321	321321	
Excluir	Selecionar	66	Filipe Maia	1010101010	8409238498	barbosa	Rui barbosa	33243170	787654365	
Excluir	Selecionar	68	Fernando Lima	123423	12341234	Professor	rua que sobe	1234213	12341234	

Figure 20: Tela de cadastro de cliente.

## Tela de agendamento

A figura abaixo apresenta a tela de agendamento de cliente onde terá todos os dados necessários para realizar um agendamento.

Easy Agenda Sair Login

Cadastrar ▾ Agendar ▾ Visualizar ▾

### Agendar Cliente

Profissional:

Cliente:

Tipo de Atendimento:

Observação:

Departamento:

Horario do gendamento:

Calendário

outubro de 2013						
setembro						novembro
dom	seg	ter	qua	qui	sex	sáb
29	30	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31	1	2
3	4	5	6	7	8	9

	Id	IdProfissional	IdCliente	Atendimento	DataHora	Observacao	Ativo	DataC
Excluir	79	12	62		29/10/2013 00:00:00	Agendamento Urgente	<input checked="" type="checkbox"/>	

Figure 21: Agendamento de Cliente.