



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

ALEXANDRE AMPUDIA FARIA

SISTEMA WEB DE GESTÃO PARA PRONTO ATENDIMENTO

ASSIS/SP

2019

ALEXANDRE AMPUDIA FARIA

SISTEMA WEB DE GESTÃO PARA PRONTO ATENDIMENTO

Trabalho de Conclusão de Curso apresentado ao Instituto Municipal de Ensino Superior de Assis, como requisito do Curso Superior de Tecnologia em Análise e desenvolvimento de Sistemas.

Orientando: Alexandre Ampudia Faria

Orientador: Dr. Almir Rogério Camolesi

ASSIS/SP

2019

FICHA CATALOGRÁFICA

F224s FARIA, Alexandre Ampudia
Sistema web de gestão para pronto atendimento /
Alexandre Ampudia Faria. – Assis, 2019.

41p.

Trabalho de conclusão do curso (Análise e Desenvolvimento de
Sistemas). – Fundação Educacional do Município de Assis-
FEMA

Orientador: Dr. Almir Rogério Camolesi

1.Web 2.PHP 3.Gestão

CDD 005.4

SISTEMA WEB DE GESTÃO PARA PRONTO ATENDIMENTO

ALEXANDRE AMPUDIA FARIA

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

Orientador: _____

Dr. Almir Rogério Camolesi

Examinador: _____

MSc. Guilherme de Cleve Farto

ASSIS

2019

DEDICATÓRIA

Dedico este trabalho aos meus pais Reinaldo e Daniela, que sempre acreditaram nos meus sonhos e sempre estiveram do meu lado.

AGRADECIMENTOS

Em primeiro lugar a DEUS, por me permitir ter essa experiência em minha vida. Aos meus familiares e amigos que me apoiaram durante todo esse período de desenvolvimento, entre eles Sergio Faria, Edvaldo Sotana.

Aos professores e colegas da instituição participaram para a conclusão deste sonho.

“Excelência é fazer uma coisa comum de um modo extraordinário.”

Booker T. Washington

RESUMO

O tema proposto neste trabalho será a automatização de um processo de um hospital de pronto atendimento situado na cidade de Maracaí, no interior de São Paulo, uma entidade sem fins lucrativos que nos dias atuais ainda executa os processos de atendimentos manualmente. O projeto foi idealizado com o objetivo de facilitar a execução, criar uma base de dados com informações completas e seguras, gerar agilidade nos atendimentos, gerar uma experiência ao paciente na forma como ele é atendido e reduzir o uso de suprimentos. O sistema possui cadastros para organizar as informações contidas no processo, área para gerar consultas e relatórios. Com a implementação do sistema, o usuário encontrara as informações importantes com facilidade e integridade.

Na elaboração do sistema, tecnologias de desenvolvimento web foram usadas como PHP, HTML 5, CSS 3, BOOTSTRAP, SCRIPT JAVA, MD5, MYSQL e SUBLIME TEXT 3.0 IDE, que serão apresentadas no artigo.

Palavras-chave: Pronto Atendimento – Gestão – Sistema.

ABSTRACT

The theme proposed in this paper will be the automation of a process of a emergency hospital located in the city of Maracaí, in the interior of São Paulo, a nonprofit entity that currently still performs the care processes manually. The project was designed with the goal of facilitating execution, creating a database with complete and secure information, generating speed of care, generating a patient experience in how it is served and reducing the use of supplies. The system has records to organize the information contained in the process, area to generate queries and reports. With the implementation of the system, the user will find the important information with ease and integrity.

In the elaboration of the system, web development technologies were used as PHP, HTML 5, CSS 3, BOOTSTRAP, SCRIPT JAVA, MD5, MYSQL and SUBLIME TEXT 3.0 IDE, which will be presented in the article.

Keywords: Emergency Care - Management - System.

LISTA DE FIGURAS

Figura 1 – Estrutura Analítica do Projeto.....	15
Figura 2 - Arquitetura do Sistema.....	16
Figura 3 - Mapa Mental	18
Figura 4 - Manter Paciente	20
Figura 5 - Realizar atendimento	21
Figura 6 - Gerar Atendimento.....	22
Figura 7 - Manter Médicos	23
Figura 8 - Manter Procedimentos	24
Figura 9 - Manter Usuários.....	25
Figura 10 - Relatórios de Atendimentos	26
Figura 11 - Diagrama de Caso de Uso Geral	28
Figura 12 - Diagrama de Classes.....	29
Figura 13 - Diagrama de Entidade e Relacionamento.....	30
Figura 14 - Login	33
Figura 15 - Código Login.....	34
Figura 16 - Tela Cadastrar Paciente	34
Figura 17- Código Cadastrar Paciente	35
Figura 18 - Tela Gerar Atendimento.....	35
Figura 19 - Código Gerar Atendimento	36
Figura 20 - Home Page	36
Figura 21 - Tela Manter Pacientes	37
Figura 22 - Tela Atendimentos em aberto	37
Figura 23 - Tela Atendimentos Realizados	38
Figura 24 - Tela de Histórico do Paciente	38
Figura 25 - Tela Manter Médicos.....	39
Figura 26 - Tela de Cadastrar Usuário	39

SUMÁRIO

1. INTRODUÇÃO	13
2. ANÁLISE DO PROJETO	15
2.1. PLANEJAMENTO DO PROJETO - EAP (Estrutura Analítica do Projeto).....	15
2.2. ARQUITETURA DO SISTEMA	16
2.3. REQUISITOS	17
2.4. BANCO DE DADOS.....	17
2.5. INTERFACES DO SISTEMA	17
2.6. SEGURANÇA	17
2.7. TESTES	18
2.8. MAPA MENTAL	18
2.9. LISTA DE EVENTOS	19
2.10. ESPECIFICAÇÃO DOS CASOS DE USO	19
2.10.1. Manter pacientes.....	19
2.10.2. Realizar atendimento.....	21
2.10.3. Gerar Atendimento	22
2.10.4. Manter Médicos.....	23
2.10.5. Manter Procedimentos	24
2.10.6. Manter Usuários.....	25
2.10.7. Relatório de Atendimentos	26
3. FERRAMENTAS E TECNOLOGIAS UTILIZADAS	28
3.1 ANÁLISE DO PROJETO.....	28
3.1.1 Diagrama de casos de uso (UC)	28
3.1.2 Diagrama de classe	28
3.1.3 Diagrama entidade e relacionamento (DER)	29
3.2 DENSENVOLVIMENTO DO SISTEMA.....	30

3.2.1 Html 5.....	30
3.2.2 Css 3	30
3.2.3 Bootstrap.....	31
3.2.4 Font awesome.....	31
3.2.5 Sublime text 3.0.....	31
3.2.6 Php.....	31
3.2.7 Mysql.....	32
3.2.8 Xampp.....	32
3.2.9 Javascript	32
3.2.10 Md5.....	33
3.3 INTERFACES DO SISTEMA	33
3.3.1. Login	33
3.3.2. Cadastrar Novo Paciente.....	34
3.3.3. Gerar Atendimento.....	35
3.3.4. Home Page.....	36
3.3.5. Manter Pacientes.....	36
3.3.6. Atendimentos em Aberto	37
3.3.7. Atendimentos Realizados.....	37
3.3.8. Histórico do Paciente	38
3.3.9. Manter Médicos	38
3.3.10. Cadastrar Usuário	39
4. CONCLUSÃO	40
5. REFERÊNCIAS.....	41

1. INTRODUÇÃO

Segundo MEHRY E et al, a industrialização trouxe consigo, além da modernização, o avanço tecnológico e a valorização da ciência em detrimento do homem e de seus valores. Os avanços tecnológicos também ocorreram na área da saúde, com a introdução da informática e do aparecimento de aparelhos modernos e sofisticados que trouxeram muitos benefícios e rapidez na luta contra as doenças. Essa tecnologia moderna, criada pelo homem a serviço do homem, tem contribuído em larga escala para a solução de problemas antes insolúveis e que pode reverter em melhores condições de vida e saúde para o paciente.

Os dias atuais caracterizam-se por profundas e constantes mudanças, onde é crescente e cada vez mais acelerada a inovação tecnológica, colocando à disposição dos profissionais e usuários, os mais diversos tipos de tecnologia, tais como: tecnologias educacionais, tecnologias gerenciais e tecnologias assistenciais.

Contudo, vivemos numa era tecnológica onde muitas vezes a concepção do termo tecnologia tem sido utilizada de forma enfática, incisiva e determinante, porém equivocada em nossa prática diária, uma vez que tem sido concebida, corriqueiramente, somente como um produto ou equipamento. A temática tecnologia não deve ser tratada através de uma concepção reducionista ou simplista, associada somente à máquinas. Entendemos que a tecnologia compreende certos saberes constituídos para a geração e utilização de produtos e para organizar as relações humanas (MEHRY E et al, 1997).

O presente trabalho foi elaborado a partir da observação de necessidade de automatização no processo de atendimento no Hospital Beneficente de Maracaí, uma entidade sem fins lucrativos, que se mantém nos dias atuais através de subvenções do SUS (Sistema Único de Saúde) da Prefeitura Municipal e de doações privadas arrecadadas por movimentos beneficentes.

Atualmente o processo de atendimento é realizado através de fichas de formulários que são preenchidos a mão pelos profissionais da entidade, desta forma, enxerga-se a necessidade de automatização da área, com o intuito de garantir a integridade e

agilidade dos atendimentos, a organização das informações e a criação de uma base de dados segura para cada paciente, de forma atualizada e de fácil acesso.

Segundo o IBGE, a população estimada para o ano de 2017 no município de Maracá-SP, fica em torno de 13.981 habitantes, dos quais a maioria é dependente do Sistema Público de Saúde. A proposta assistencial da entidade está voltada para o atendimento de pacientes que utilizam os serviços do SUS. Apesar de haver pouca complexidade dos serviços prestados, sendo um hospital de pequeno porte, a entidade tem potencial significativo de resolutividade para atendimentos de urgência - emergência.

Atendendo as regras de funcionamento da entidade, foi desenvolvido um projeto junto aos colaboradores, para verificar falhas e possibilidades de melhoria nos atendimentos aos usuários afim de tornar a assistência mais rápida e eficaz. Juntamente ao sistema de informação, foi implementada a ideia de gestão sustentável, que consiste em reduzir o uso de papeis, equipamentos e suprimentos necessários para a atual realização do processo. Com essas medidas espera-se um corte de gastos considerável para a entidade e agilidade nos processos de atendimento aos pacientes.

2. ANÁLISE DO PROJETO

2.1. PLANEJAMENTO DO PROJETO - EAP (Estrutura Analítica do Projeto)

A Estrutura Analítica do Projeto (EAP) é utilizada em projetos para organizar e confirmar o escopo total do projeto a ser desenvolvido. De acordo com o PMI, a EAP evita que alguma atividade que precise ser executada, seja esquecida e também detalha todo o trabalho definido para na declaração do escopo.

O projeto foi elaborado com o intuito de centralizar as informações principais do processo de atendimento de forma segura, onde organizamos os cadastros, os relatórios e a forma como são gerados os atendimentos e buscando tornar acessível o uso da tecnologia no dia a dia da instituição, junto a ideia de automatização inserimos o conceito de sustentabilidade, reduzindo o uso de recursos e materiais utilizados no processo atual. A figura abaixo representa a Estrutura Analítica do Projeto.

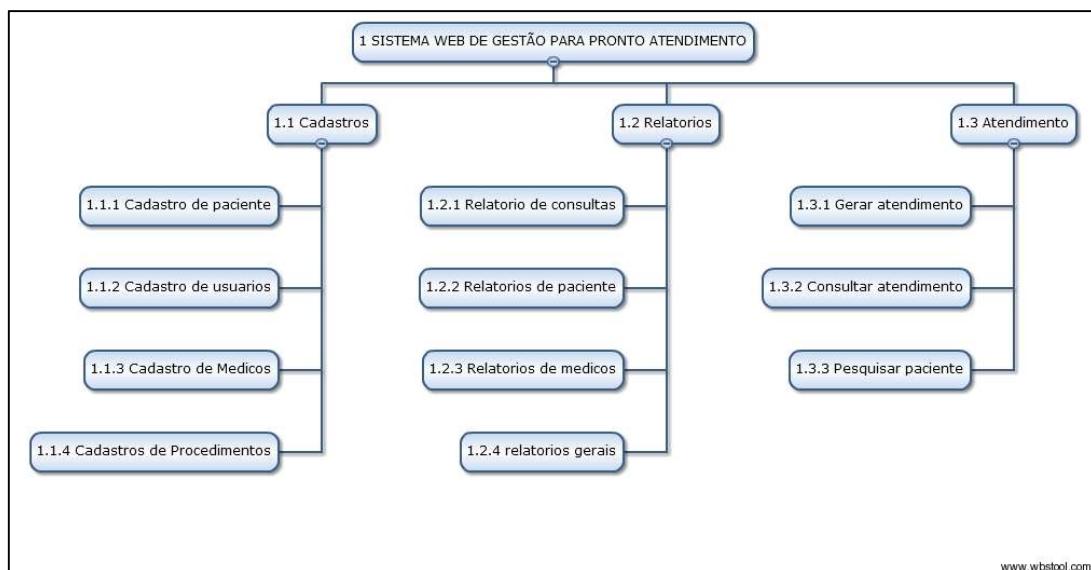


Figura 1 – Estrutura Analítica do Projeto

Partindo do conceito do projeto, foram definidas as seguintes etapas de execução conforme Figura 1:

- Levantamento de requisitos;
- Elaboração do banco de dados;

- Criação das tabelas principais;
- Estudos e implementação de tecnologias para criação e formatação;
- Implementação das telas do sistema;
- Estudos e implementação de segurança;
- Testes finais.

2.2. ARQUITETURA DO SISTEMA

Após análise da necessidade do projeto, foi selecionada a arquitetura cliente-servidor que consiste em processos executados apenas no servidor, onde o cliente realiza uma requisição através do protocolo HTTP (Hypertext Transfer Protocol) que é transmitido através da web e enviada ao servidor, após a execução do processo solicitado, o servidor realiza uma consulta ao banco de dados, onde os dados são processados e tratados e em sequência o servidor realiza uma resposta através de uma página HTML, em conjunto com CSS, BOOTSTRAP e JAVASCRIPT, onde os dados retornados do servidor são apresentados em telas para o cliente. Sendo assim, o usuário final não tem acesso ao servidor, ele apenas recebe páginas com dados inseridos. A figura abaixo representa a arquitetura cliente servidor que foi implementada no projeto

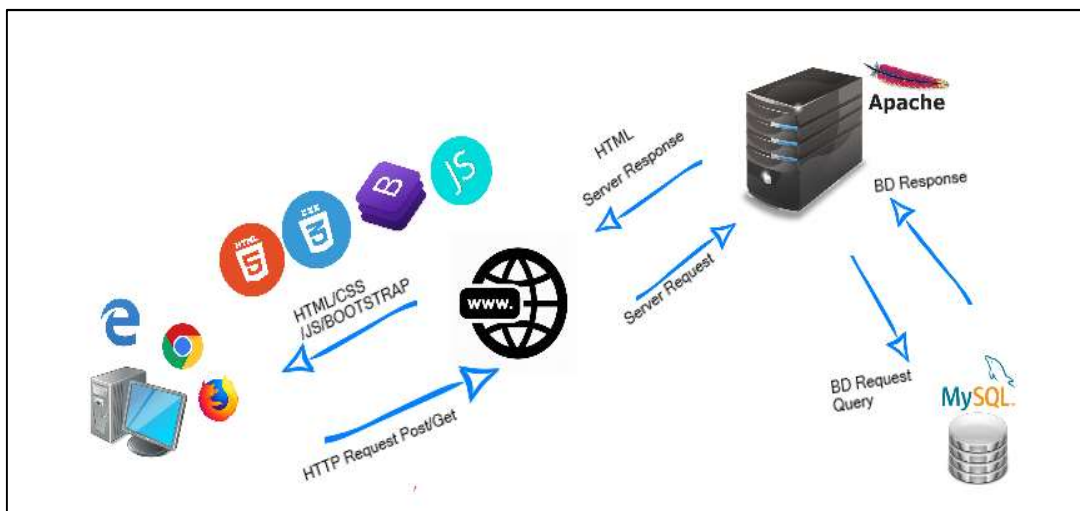


Figura 2 - Arquitetura do Sistema

2.3. REQUISITOS

Na etapa inicial do desenvolvimento, foi efetuada uma entrevista com os profissionais da área de saúde da entidade questionando qual a maior necessidade em suas atividades diárias relacionadas à tecnologia. Em seguida junto ao administrador da entidade foi realizado o levantamento de requisitos para implementação sistêmica do projeto, onde buscamos informações que julgamos relevantes para formular informações consistentes e validas.

2.4. BANCO DE DADOS

Analisando os requisitos levantados na entrevista, foi realizado a implementação das principais tabelas no banco de dados, onde foram incluídas tabelas de cadastros e usuários. Consideramos essa como uma etapa importante para o desenvolvimento do projeto, onde a estrutura deve ser elaborada de maneira clara, para não gerar transtornos nas etapas subsequentes.

2.5. INTERFACES DO SISTEMA

Em conjunto aos diretores da entidade foi elaborada uma escala de cores padrão para o sistema levando em consideração o logotipo da entidade e as cores padrões utilizadas nas bandeiras, onde foi selecionada a cor azul (Escala RGB: #428bca) para ser predominante nas telas. Os ícones foram definidos de acordo com as seções que foram implementadas, onde utilizamos o framework font awesome que dispõe de uma ampla biblioteca de ícones. Foi utilizado os conceitos de CSS (Cascading Style Sheets) na formatação dos botões, títulos, layout dos formulários e posições.

Na tela de login do sistema foi utilizado um template editável gratuito, de domínio público, disponível no portal Bootsnipp, o qual foi adaptado de acordo com os padrões do projeto.

2.6. SEGURANÇA

Segundo o portal DevMedia criptografia é o ato de codificar dados em informações aparentemente sem sentido, para que pessoas não consigam ter acesso às informações que foram cifradas. Foi implementado no sistema a criptografia MD5,

ferramenta de domínio público que consiste em um algoritmo unidirecional, portanto não é possível ser revertido após a execução. A verificação é realizada através de um hash, onde os dados armazenados no banco são comparados ao cálculo efetuado pelo algoritmo

2.7. TESTES

Após o desenvolvimento do sistema, foram efetuados testes nos formulários, buscando verificar se o sistema estava consistente e seguro no momento da execução dos procedimentos. Esta etapa foi relevante para que o sistema não apresentasse falhas, inclusões de dados nulos e a falta de informações relevantes.

2.8. MAPA MENTAL

O Mapa Mental é uma forma objetiva e simples de registrar informações. Essa técnica, que é uma espécie de diagrama, sistematizado pelo psicólogo inglês Tony Buzan, é muito usada para anotação de conteúdo e está relacionada à gestão de informações, ao conhecimento e ao capital intelectual, a fim de solucionar problemas e atender aos requisitos de memorização e aprendizado.

Para tornar o entendimento claro do sistema, foi desenvolvido um mapa mental das funcionalidades do sistema, conforme a figura 2. Para a criação do Mapa Mental, foi utilizada a ferramenta Coggle (ferramenta disponível em: <https://coggle.it/>), é uma ferramenta gratuita de domínio que possui uma maneira dinâmica e simples para organizar as informações. A figura a seguir representa o mapa mental do projeto.

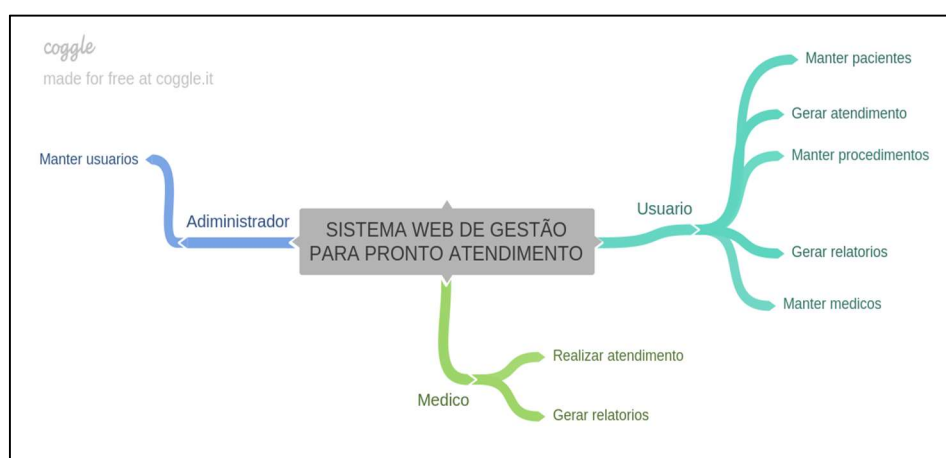


Figura 3 - Mapa Mental

2.9. LISTA DE EVENTOS

A lista de eventos representa os atores responsáveis por cada ação que ocorre no sistema e a descrição de funcionalidades implementadas. A tabela a seguir representa a lista de eventos do sistema.

Lista de Eventos			
Nº	Evento	Descrição	Ator
1	Manter médicos	Manter médicos no sistema	Usuário
2	Manter procedimentos	Manter procedimentos no sistema	Usuário
3	Manter usuários	Manter usuários no sistema	Administrador
4	Manter consultas	Manter consultas no sistema	Usuário
5	Gerar relatórios	Gerar relatórios no sistema	Administrador
6	Gerar atendimentos	Gerar atendimentos no sistema	Usuário
7	Manter Pacientes	Manter pacientes no sistema	Usuário
8	Gerar relatórios	Gerar relatórios no sistema	Usuário
9	Realizar atendimento	Realizar atendimentos no sistema	Medico
10	Gerar relatório	Gerar relatórios no sistema	Medico

Tabela 1 –Lista de Eventos

2.10. ESPECIFICAÇÃO DOS CASOS DE USO

2.10.1. Manter pacientes

A figura a seguir representa o caso de uso manter pacientes que permite ao ator cadastrar novos pacientes que não estejam na base de dados, editar dados de um paciente já cadastrado e permite realizar uma busca de um paciente pelo número do cartão sus.

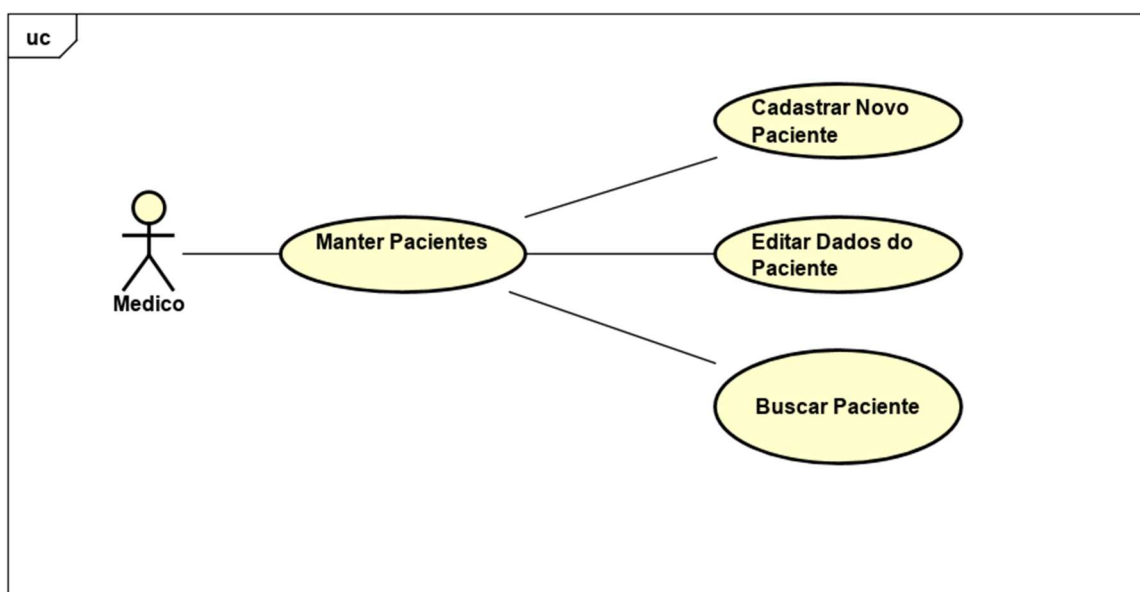


Figura 4 - Manter Paciente

Nome da Use Case (CSU01)	Manter Paciente
Ator (es)	Usuário
Pré-Condições	Cadastrar um paciente ainda não cadastrado
Cenário Principal	<ol style="list-style-type: none"> 1. O paciente deseja realizar o cadastro no sistema. 2. O usuário acessa o menu Adicionar novo paciente. (A1) 3. O paciente informa os dados pessoais que serão solicitados pelo sistema. 4. O usuário confirma os dados. 5. O sistema cadastra o paciente na base de dados.
Cenário Alternativo – A1	O usuário poderá inativar um cliente já cadastrado.

Tabela 2 – Manter Paciente

2.10.2. Realizar atendimento

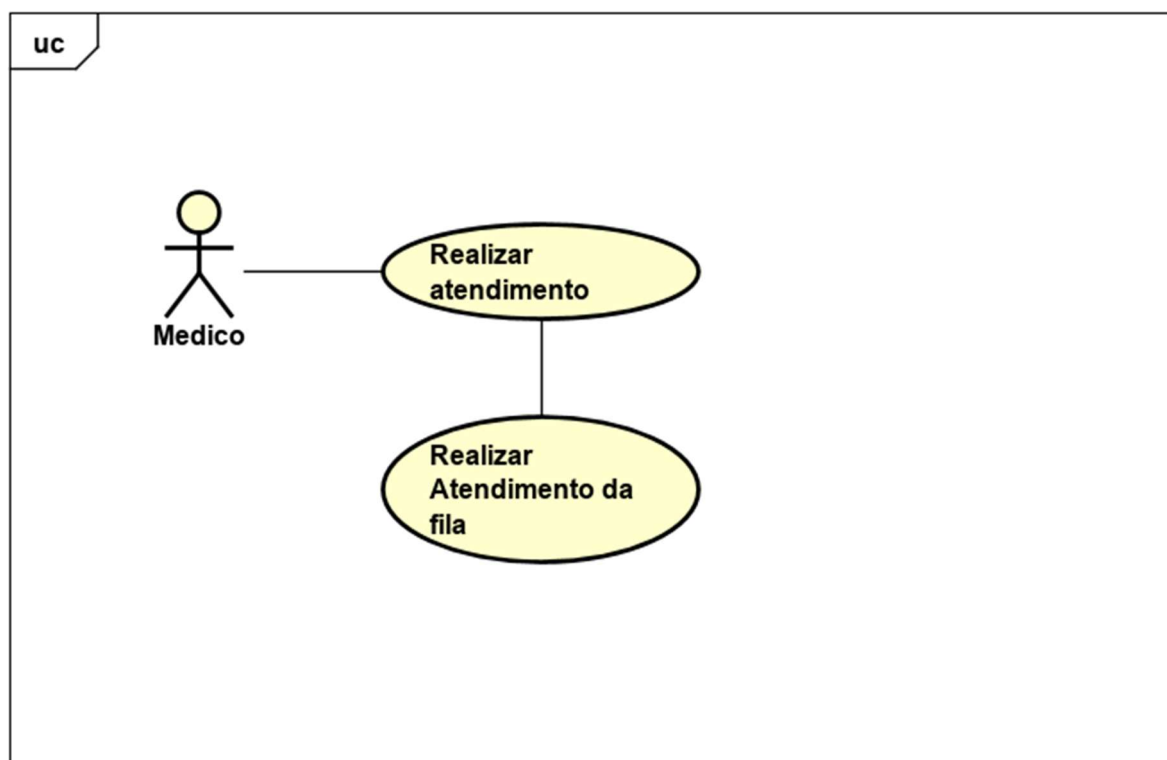


Figura 5 - Realizar atendimento

Nome da Use Case (CSU02)	Realizar atendimento
Ator (es)	Medico
Pré-Condições	Realizar um atendimento ainda não realizado
Cenário Principal	<ol style="list-style-type: none"> 1. Um novo atendimento é gerado na área de atendimentos. 2. O médico acessa a aba de realizar atendimentos. 3. O médico verifica as informações informadas pelo sistema e realiza a chamada do paciente. 4. O médico realiza atendimento e insere os dados necessários.

	<p>5. O Médico confirma o atendimento.</p> <p>6. O sistema registra o atendimento.</p>
--	--

Tabela 4 – Realizar atendimento

2.10.3. Gerar Atendimento

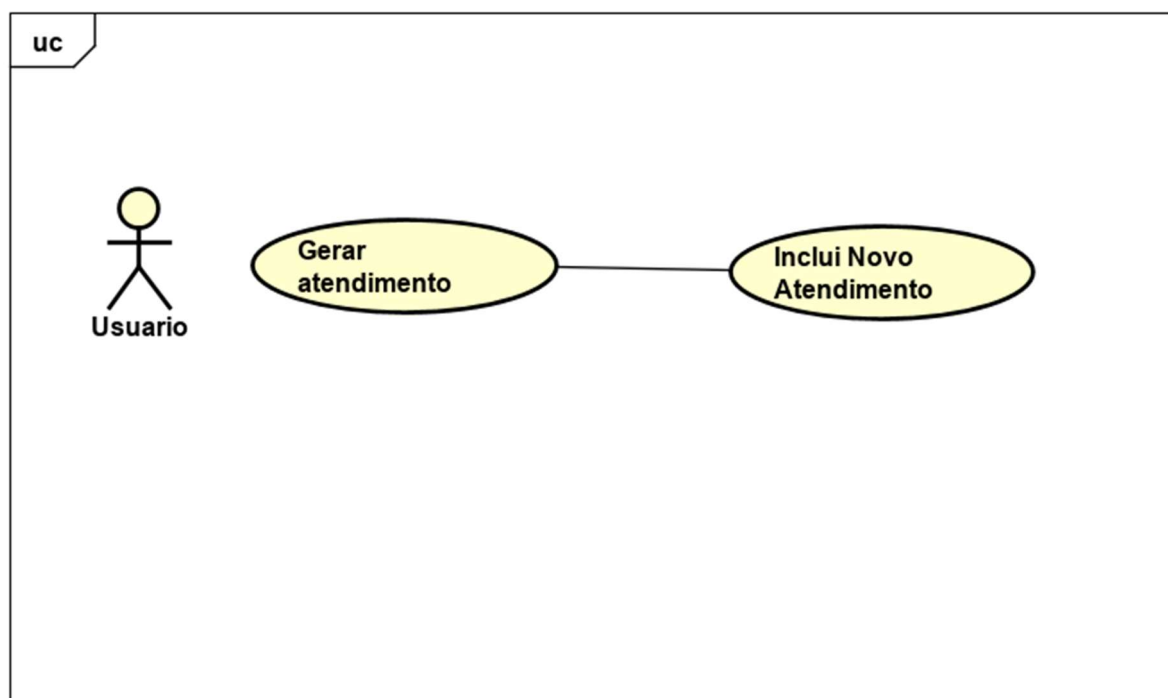


Figura 6 - Gerar Atendimento

Nome da Use Case (CSU02)	Gerar atendimento
Ator (es)	Usuário
Pré-Condições	Gerar um novo atendimento
Cenário Principal	<ol style="list-style-type: none"> 1. O usuário acessa a função gerar atendimento. 2. O sistema exibe o formulário com as informações que devem ser inseridas pelo usuário.

	<p>3. O usuário insere os dados e confirma.</p> <p>4. O sistema registra o atendimento.</p>
--	---

Tabela 5 – Gerar atendimento

2.10.4. Manter Médicos

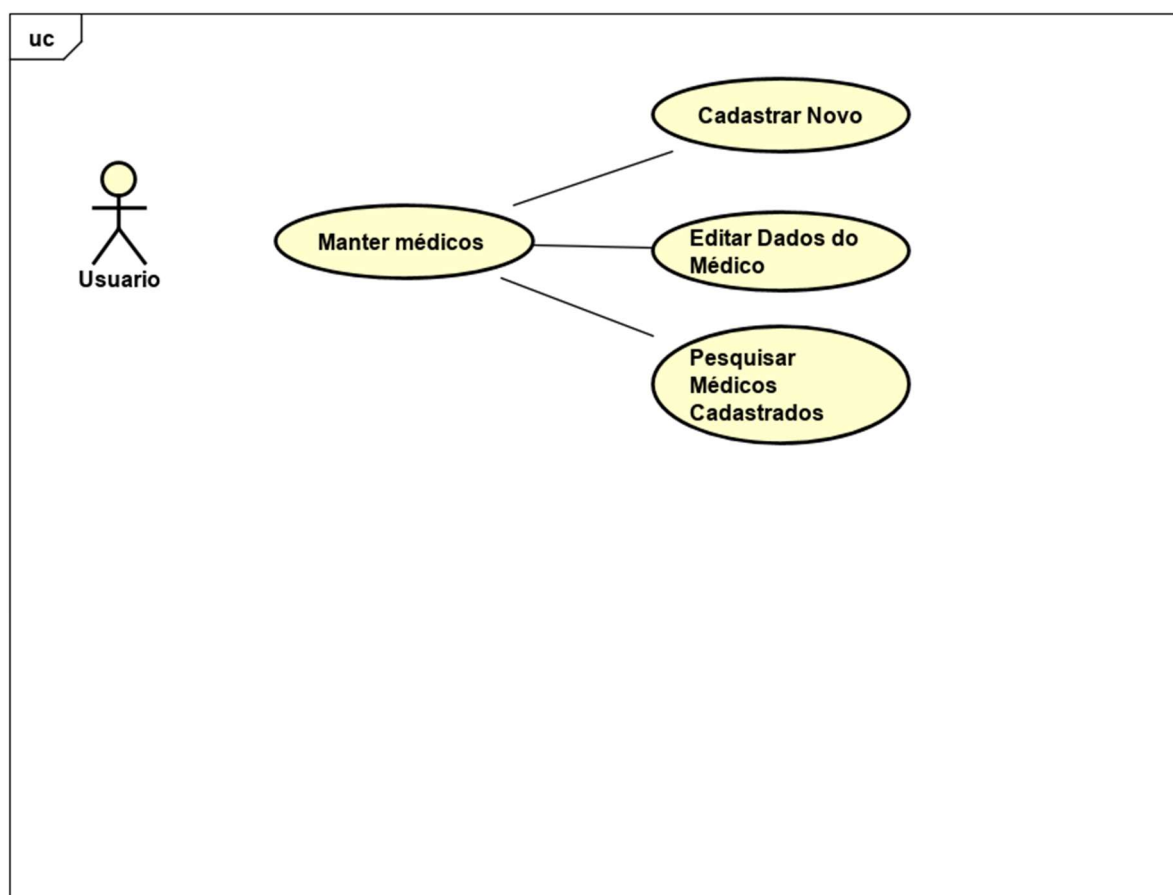


Figura 7 - Manter Médicos

Nome da Use Case (CSU02)	Manter Médicos
Ator (es)	Usuário
Pré-Condições	Efetuar Login no sistema
Cenário Principal	<p>1. O usuário acessa a função Manter Médicos.</p>

	<ol style="list-style-type: none"> 2. O sistema exibe o formulário com as informações da lista de médicos. 3. O usuário poderá incluir, alterar e visualizar os dados dos médicos. 4. O sistema registra a ação executada.
--	---

Tabela 6 – Manter Médicos

2.10.5. Manter Procedimentos

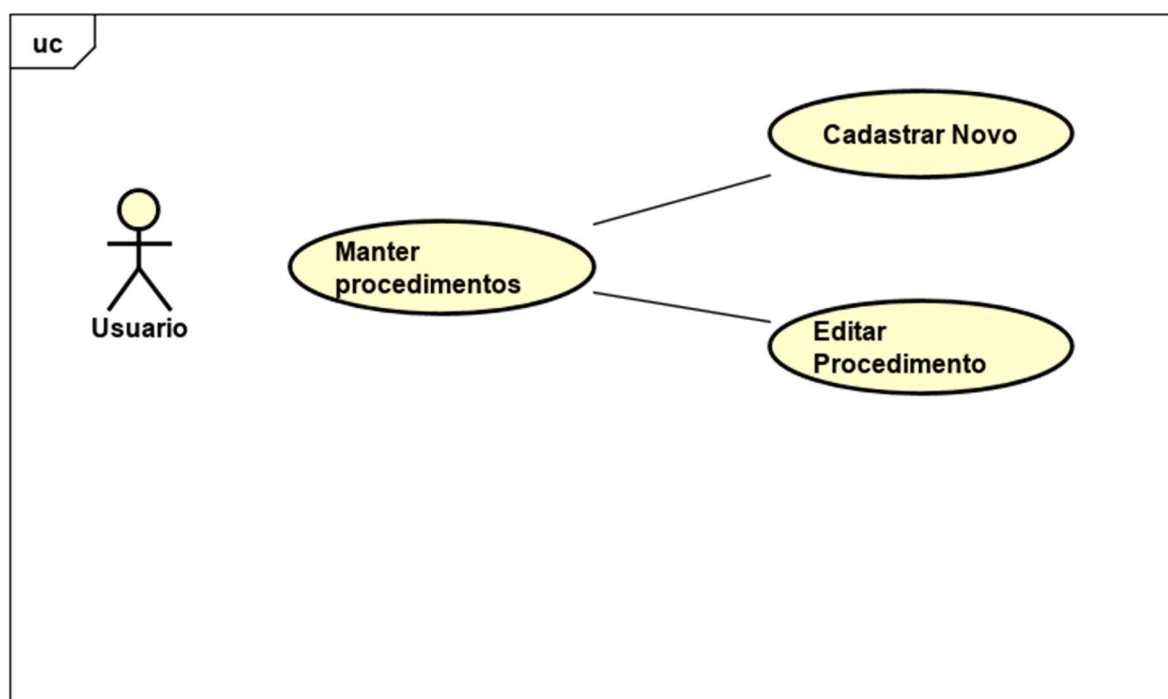


Figura 8 - Manter Procedimentos

Nome da Use Case (CSU02)	Manter Procedimentos
Ator (es)	Usuário
Pré-Condições	Efetuar Login no sistema
Cenário Principal	<ol style="list-style-type: none"> 1. O usuário acessa a função Manter Procedimentos.

	<ol style="list-style-type: none"> 2. O sistema exibe o formulário com as informações da lista de procedimentos. 3. O usuário poderá incluir, alterar e visualizar os dados dos procedimentos. 4. O sistema registra a ação executada.
--	---

Tabela 7 – Manter Procedimentos

2.10.6. Manter Usuários

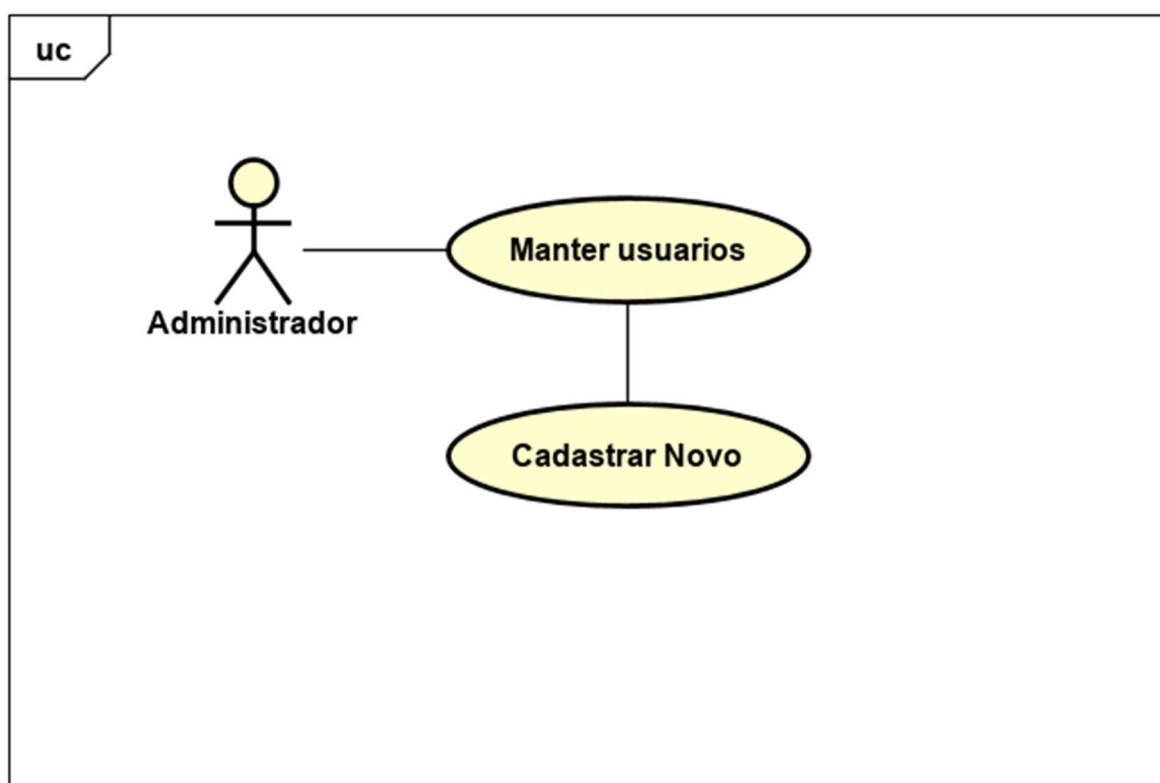


Figura 9 - Manter Usuários

Nome da Use Case (CSU02)	Manter Usuários
Ator (es)	Administrador
Pré-Condições	Efetuar Login no sistema

<p>Cenário Principal</p>	<ol style="list-style-type: none"> 1. O usuário acessa a função Manter Usuários. 2. O sistema exibe o formulário com as informações necessárias para incluir um novo usuário. 3. O usuário insere os dados solicitados e confirma. 4. O sistema registra a ação executada.
---------------------------------	--

Tabela 8 – Manter Usuários

2.10.7. Relatório de Atendimentos

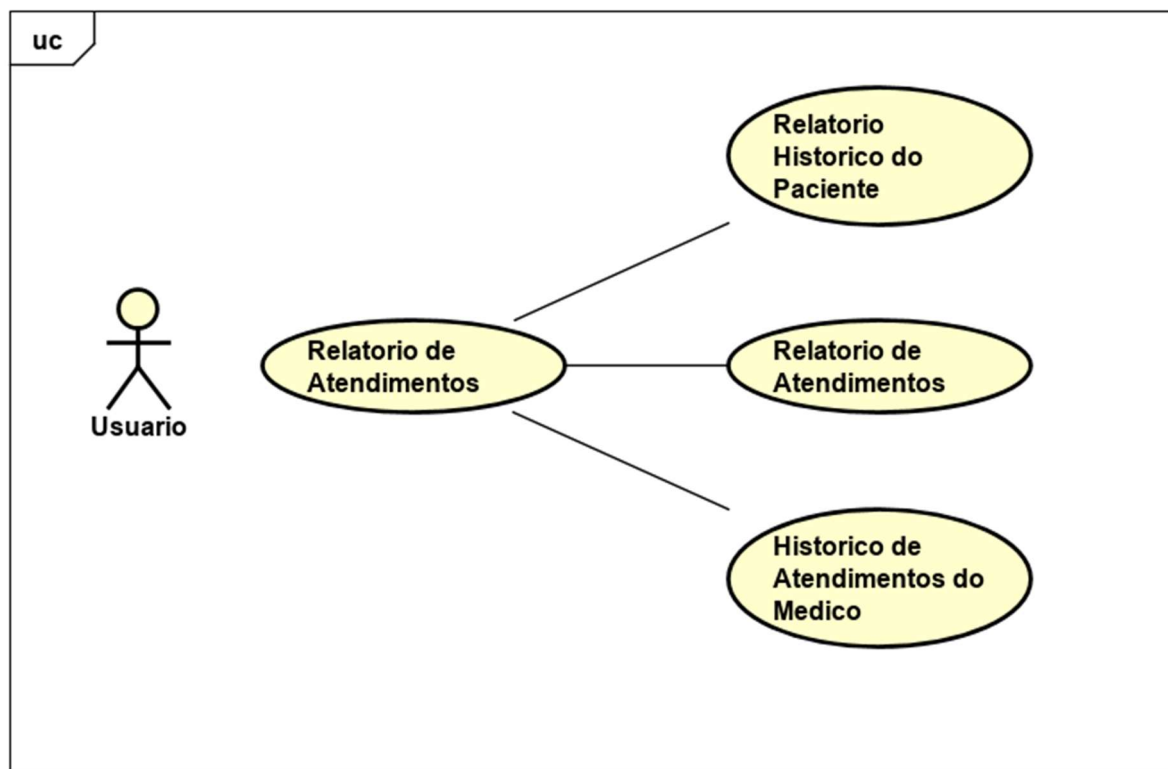


Figura 10 - Relatórios de Atendimentos

Nome da Use Case (CSU02)	Relatório
Ator (es)	Usuário
Pré-Condições	Efetuar Login no sistema

Cenário Principal	<ol style="list-style-type: none">1. O usuário acessa a função Relatórios.2. O sistema exibe o formulário com as informações.3. O sistema possibilita a impressão do relatório gerado
--------------------------	---

Tabela 9 – Relatórios de Atendimentos

3. FERRAMENTAS E TECNOLOGIAS UTILIZADAS

3.1 ANÁLISE DO PROJETO

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

3.1.1 Diagrama de casos de uso (UC)

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

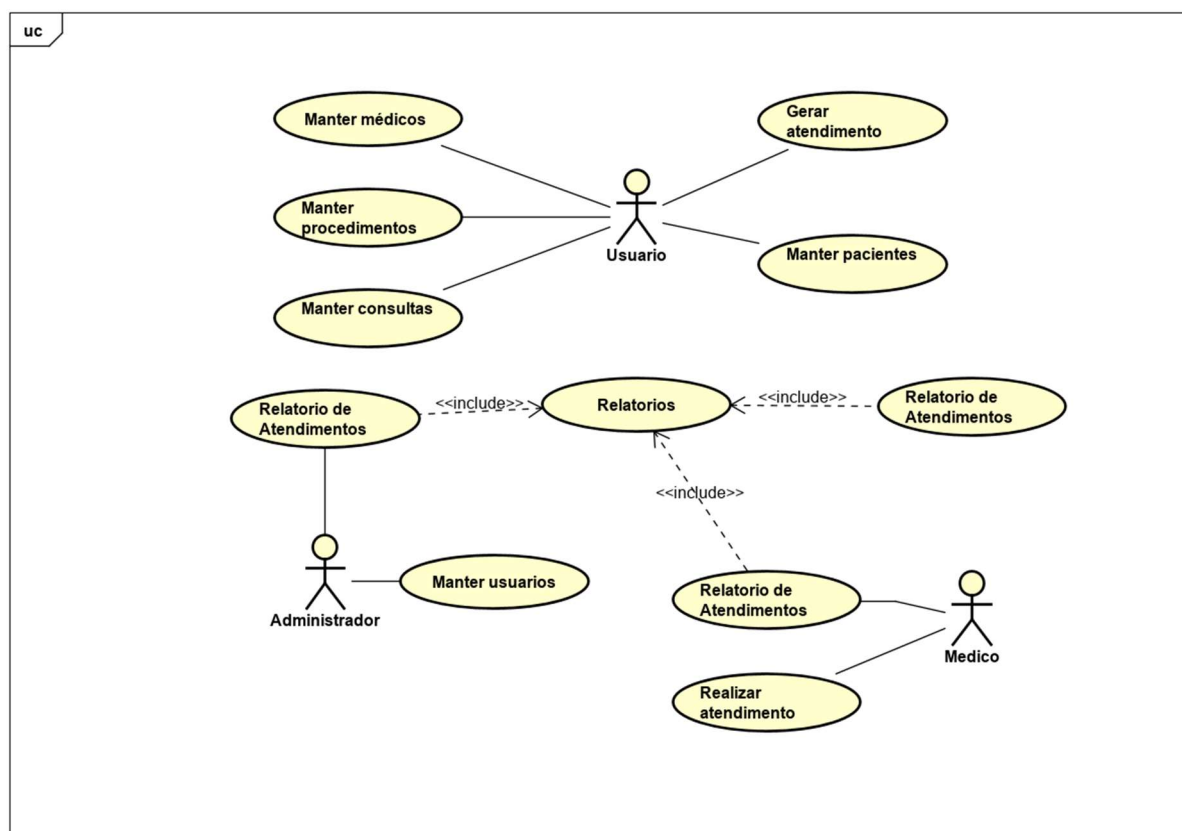


Figura 11 - Diagrama de Caso de Uso Geral

3.1.2 Diagrama de classe

O diagrama de classe ilustra um conjunto de classes e seus relacionamentos de forma estática de um sistema, eles são importantes não só para documentação, mas também para o desenvolvimento do sistema contribuindo de forma impactante para a

consolidação do escopo das funcionalidades do projeto. (GÓES, 2014). A figura abaixo representa o diagrama de classes do sistema.

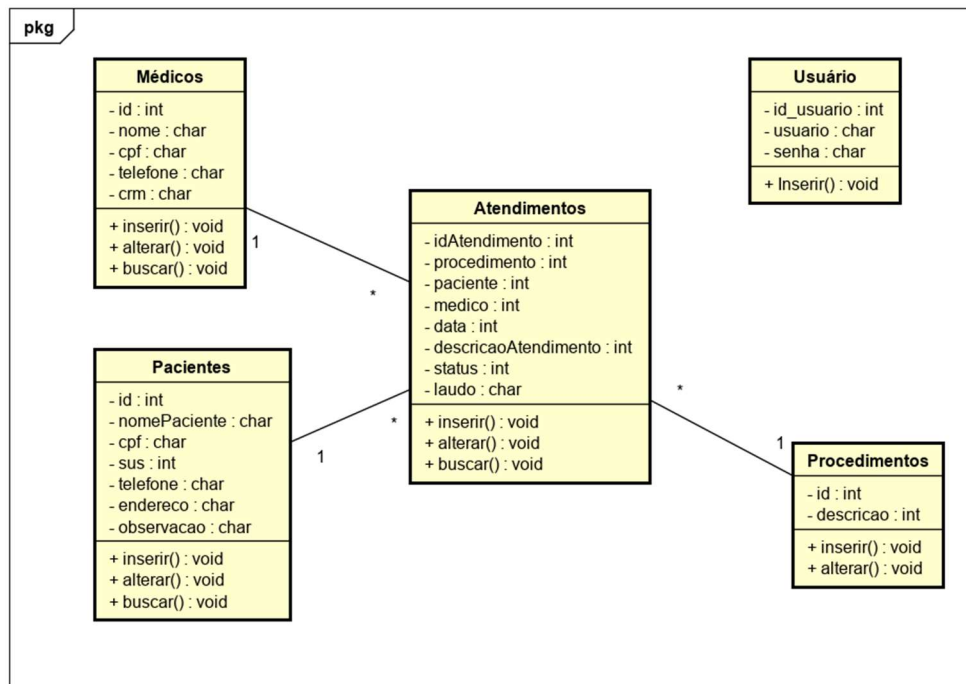


Figura 12 - Diagrama de Classes

3.1.3 Diagrama entidade e relacionamento (DER)

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

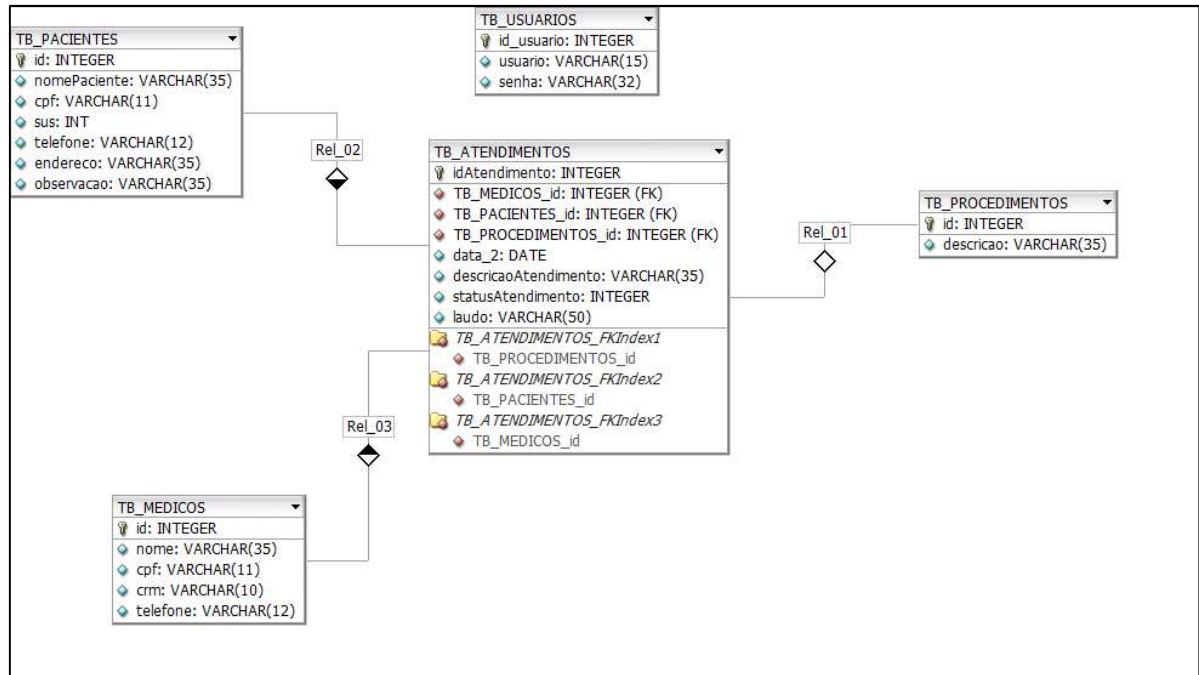


Figura 13 - Diagrama de Entidade e Relacionamento

3.2 DENSENVOLVIMENTO DO SISTEMA

3.2.1 Html 5

O HTML é uma linguagem muito utilizada na criação de páginas web, onde a sigla é uma abreviação do nome Hypertext Markup Language. Foi criada com o objetivo de trazer a compatibilidade com diferentes plataformas de navegação para gerar publicações de fotos, vídeos, e conteúdos de texto. Utilizamos essa linguagem no front-end onde tratamos o visual da página web, no projeto foi implementado a versão 5 do HTML, versão que trás atualizações em relação as versões mais antigas. (MILANI, André. **MySQL-guia do programador**. Novatec Editora, 2007.)

3.2.2 Css 3

A linguagem utilizada para formatação criação de estilos do visual de páginas web é chamada de CSS, que é uma abreviação do nome Cascading Style Sheets, linguagem que surgiu com a necessidade de padronizar os métodos de criar formatações e estilos aplicados nas páginas web. Ao longo do tempo essa linguagem passou por atualizações e implementamos no projeto a versão 3 do CSS. (SILVA, Maurício Samy. **CSS3: desenvolva aplicações web profissionais com uso dos poderosos recursos de estilização das CSS3**. Novatec Editora, 2011.)

3.2.3 Bootstrap

O framework bootstrap é uma ferramenta utilizada para criar interfaces e layouts que podem ser padronizados e personalizados de acordo com a necessidade do projeto, atua em conjunto com o HTML, CSS e JavaScript. Ferramenta utilizada no desenvolvimento do front-end do projeto utilizando a versão 4.3.x.

3.2.4 Font awesome

Essa ferramenta é utilizada na implementação de fontes e ícones de maneira dinâmica nas páginas web, podendo ser incluída junto ao projeto e também ser utilizada de forma online através de um link, atuando em conjunto com o HTML e CSS na criação do front-end das páginas. No projeto utilizamos a versão online da ferramenta onde o download dos ícones é feito através do link.

3.2.5 Sublime text 3.0

O Sublime é um programa multiplataforma de edição de texto muito utilizado para linguagens como PHP, PYTHON. O sublime muitas vezes é comparado a uma IDE (Integrated Development Environment), mas é somente um editor de texto.

3.2.6 Php

O PHP é uma linguagem utilizada no desenvolvimento de sites dinâmicos, onde existe possibilidade de criação de formulários, execução de diversos processos em backend. É uma linguagem interpretada pelo servidor, que ao ser integrada com tecnologias de desenvolvimento de front-end, executa os processos implementados de forma segura e eficaz.

Segundo a revista EXAME, o PHP está no ranking das linguagens de desenvolvimento web mais utilizadas da atualidade, por ser uma linguagem agradável e popular. Uma das vantagens de trabalhar com uma linguagem popular é porque isso torna mais rico o número de recursos disponíveis, e também a quantidade de pessoas dispostas a ajudar, e claro, conseqüentemente o número de tutoriais e artigos disponíveis é outro fator excelente para a linguagem.

Embora o PHP seja amplamente utilizado em projetos pequenos, como sites, blogs e etc, é uma linguagem muito utilizada em grandes empresas, inclusive multinacionais, algumas empresas utilizam a linguagem como base para seus produtos externos, e

em outros casos a linguagem é utilizada para o desenvolvimento de funcionalidades internas, como intranets por exemplo.

Nesse projeto foi implementado tarefas de acesso a banco de dados, inclusões e tratamento de dados, formas de segurança e validação utilizando a linguagem.

3.2.7 Mysql

Segundo o estudo realizado pelo Ranking DB-Engines, o MySQL está entre os primeiros colocados no ranking dos bancos de dados mais utilizados mundialmente. O MySQL é um sistema gerenciador de banco de dados desenvolvido pela MySQL AB, faz parte da Oracle Corporation. É composto pelo modelo relacional que tem como principal função estruturar e armazenar informações.

3.2.8 Xampp

Segundo João Paulo Cercal do portal cercal.io, a ferramenta XAMPP é um pacote de multi ferramentas onde as principais ferramentas são, o servidor apache, o gerenciador de banco de dados MySQL e o PHPMyAdmin e o servidor FTP. Compatível com todos os sistemas operacionais atualmente é muito usado nos estudos para desenvolvimento web

Implementamos no sistema os conceitos de banco de dados utilizando o MySQL, em razão da forma dinâmica e rápida que o sistema oferece de gerenciamento de estruturas e tabelas.

3.2.9 Javascript

O JavaScript é uma linguagem considerada de alto nível, que possui mais características com a linguagem humana do que em relação a linguagem de códigos que são interpretadas por computadores, consideramos também como uma linguagem dinâmica que atua em conjunto com o HTML e o CSS. O JavaScript está no ranking das 3 linguagens mais utilizadas no desenvolvimento web. O projeto implementa a biblioteca jQuery muito utilizada do JavaScript com diversas funções entre elas a validação de formulários. (SILVA, Maurício Samy. **JavaScript-Guia do Programador: Guia completo das funcionalidades de linguagem JavaScript.** Novatec Editora, 2010).

3.2.10 Md5

O MD5 é uma ferramenta utilizada no projeto para gerar a criptografia em senhas de usuários. Na função é efetuado um cálculo irreversível onde é gerado um hash que é uma comparação com os dados já arquivados para verificar se são compatíveis

3.3 INTERFACES DO SISTEMA

3.3.1. Login

O login é a tela de acesso ao sistema, onde o usuário insere as suas informações para acessar o sistema, de acordo com a figura 14.

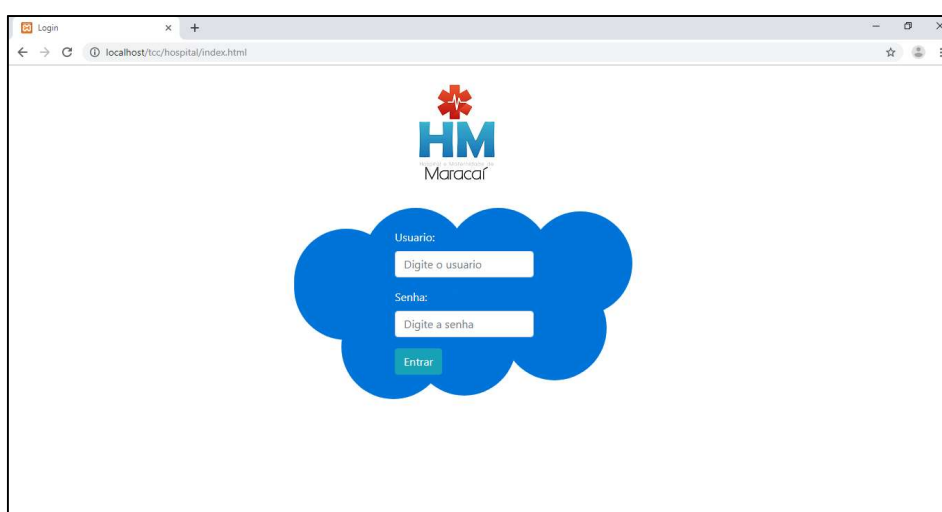


Figura 14 - Login

A figura abaixo representa o código fonte da página de login do sistema, onde é realizado o acesso ao banco de dados e ocorre a validação.

```

7     }
8
9     $banco = mysql_select_db("hospital"); // seleciona o banco a ser usado
10    if(!$banco){
11        echo "Erro ao se conectar com o banco trabalho";
12        exit;
13    }
14
15    $user = trim($_POST['usuario']);
16    $pwd = trim($_POST['senha']);
17    $pwd = md5($pwd);
18
19    $rs = mysql_query("SELECT * FROM usuarios where usuario like '$user'");
20    $linha = mysql_fetch_array($rs);
21
22    //echo ($linha['user'] . "-" . $linha['pwd']);
23
24    if ($pwd==$linha['senha']) {
25        session_start();
26        $_SESSION['usuario'] = $user;
27        header('location: home.php');
28    }
29    else
30    {
31        echo "Usuario ou senha incorretos";
32        header('location: index.html' );
33    }
34

```

Figura 15 - Código Login

3.3.2. Cadastrar Novo Paciente

A tela cadastrar novo paciente permite ao usuário realizar o cadastro de um novo paciente na base de dados, o formulário de cadastro possui validação de campos, onde os dados solicitados devem ser inseridos, caso contrário, a execução da função não ocorre, conforme a figura 16.

Figura 16 - Tela Cadastrar Paciente

A figura abaixo representa o código fonte da página de cadastro de pacientes, onde é realizado a inserção de dados no banco.

```

1 <?php
2
3 session_start();
4 if (!isset($_SESSION['usuario'])) {
5     header("Location: index.html");
6 }
7
8 $conexao = mysql_connect("localhost","root",""); //abre a conexao com banco
9
10 if (!$conexao){
11     echo "Erro ao se References:
12     exit;          atendimentosRegistrados.php:9
13                 cadastrarMedico.php:8
14                 cadastrarProcedimento.php:8
15                 cadastrarUsuario.php:8
16                 editarMedico.php:8
17                 editarPaciente.php:8
18                 fichaAtendimento.php:8
19                 fichaMedico.php:7
20                 fichaPaciente.php:7
21                 gravaAtendimento.php:7
22                 historico.php:8
23                 imprimir.php:8
24                 $nome = trim($_POST['nomePaciente']);
25                 $cpf = trim($_POST['cpf']);
26                 $sus = trim($_POST['sus']);
27                 $telefone = trim($_POST['telefone']);
28                 $endereco = trim($_POST['endereco']);
29                 $observacao = trim($_POST['observacao']);
30
31                 $sql = "INSERT INTO pacientes (cpf, nomePaciente, sus, telefone, endereco, observacao) VALUES ('$cpf','$nome','$sus', '$
32                 telefone', '$endereco', '$observacao')";
33
34                 $ins = mysql_query($sql); // comando para inserir no banco
35                 header("location:home.php");
36 }
37 ?>

```

Figura 17- Código Cadastrar Paciente

3.3.3. Gerar Atendimento

A tela gerar atendimento permite ao usuário incluir um atendimento na fila de atendimentos em aberto, e caso o paciente não possua cadastro na entidade, a tela possui um botão que permite um cadastro novo paciente no momento da inclusão do atendimento. O formulário de inclusão possui validação de campos, onde os dados solicitados devem ser inseridos, caso contrário, a execução da função não ocorre, conforme a figura 18.

Figura 18 - Tela Gerar Atendimento

A figura abaixo representa o código fonte da página de cadastro de pacientes, onde é realizado a inserção de dados no banco.

```

98 <form class="gerarAtendimento" id="myform" name="myform" method="POST"
99 action="insertAtendimento.php">
100 <div class="form-group col-md-5">
101
102 <label>Tipo de procedimento:</label>
103 <select class="custom-select" name="txtProcedimentos" id="txtProcedimentos" required="">
104 <option>Selecione</option>
105 <?php
106 $result_procedimentos = "SELECT * FROM procedimentos";
107 $resultado_procedimentos = mysqli_query($conn, $result_procedimentos);
108 while($row_procedimentos = mysqli_fetch_assoc($resultado_procedimentos)){ ??
109 <option value="<?php echo $row_procedimentos['id']; ?>"><?php echo $row_procedimentos['descricao']; ?></option>
110 <?php
111 }
112 </select>
113 <div id="campoProcedimento" style="color: red;"></div>
114 <script src="teste2.js"></script>
115 </div>
116
117 <div class="form-group col-md-5">
118 <label>Pacientes </label>
119 <select class="custom-select" name="txtPacientes" id="txtPacientes">
120 <option>Selecione</option>
121 <?php
122 $result_pacientes = "SELECT * FROM pacientes";
123 $resultado_pacientes = mysqli_query($conn, $result_pacientes);
124 while($row_pacientes = mysqli_fetch_assoc($resultado_pacientes)){ ??
125 <option value="<?php echo $row_pacientes['id']; ?>"><?php echo $row_pacientes['sus'],
126 ' - ', $row_pacientes['nomePaciente']; ?></option> <?php
127 }
128 </select>
129 <div id="campoPaciente" style="color: red;"></div>
130 <script src="teste2.js"></script>
131 </div>
132
133 <div class="form-group col-md-5">
134 <label>Pacientes </label>
135 <select class="custom-select" name="txtPacientes" id="txtPacientes">
136 <option>Selecione</option>
137 <?php
138 $result_pacientes = "SELECT * FROM pacientes";
139 $resultado_pacientes = mysqli_query($conn, $result_pacientes);
140 while($row_pacientes = mysqli_fetch_assoc($resultado_pacientes)){ ??
141 <option value="<?php echo $row_pacientes['id']; ?>"><?php echo $row_pacientes['sus'],
142 ' - ', $row_pacientes['nomePaciente']; ?></option> <?php
143 }
144 </select>
145 <div id="campoPaciente" style="color: red;"></div>
146 <script src="teste2.js"></script>
147 </div>
148
149 <div class="form-group col-md-5">
150 <label>Pacientes </label>
151 <select class="custom-select" name="txtPacientes" id="txtPacientes">
152 <option>Selecione</option>
153 <?php
154 $result_pacientes = "SELECT * FROM pacientes";
155 $resultado_pacientes = mysqli_query($conn, $result_pacientes);
156 while($row_pacientes = mysqli_fetch_assoc($resultado_pacientes)){ ??
157 <option value="<?php echo $row_pacientes['id']; ?>"><?php echo $row_pacientes['sus'],
158 ' - ', $row_pacientes['nomePaciente']; ?></option> <?php
159 }
160 </select>
161 <div id="campoPaciente" style="color: red;"></div>
162 <script src="teste2.js"></script>
163 </div>
164
165 <div class="form-group col-md-5">
166 <label>Pacientes </label>
167 <select class="custom-select" name="txtPacientes" id="txtPacientes">
168 <option>Selecione</option>
169 <?php
170 $result_pacientes = "SELECT * FROM pacientes";
171 $resultado_pacientes = mysqli_query($conn, $result_pacientes);
172 while($row_pacientes = mysqli_fetch_assoc($resultado_pacientes)){ ??
173 <option value="<?php echo $row_pacientes['id']; ?>"><?php echo $row_pacientes['sus'],
174 ' - ', $row_pacientes['nomePaciente']; ?></option> <?php
175 }
176 </select>
177 <div id="campoPaciente" style="color: red;"></div>
178 <script src="teste2.js"></script>
179 </div>

```

Figura 19 - Código Gerar Atendimento

3.3.4. Home Page

A Home Page é a tela principal do sistema, isto é, onde o usuário visualiza e acessa todas as funções principais do mesmo, conforme a figura 20.

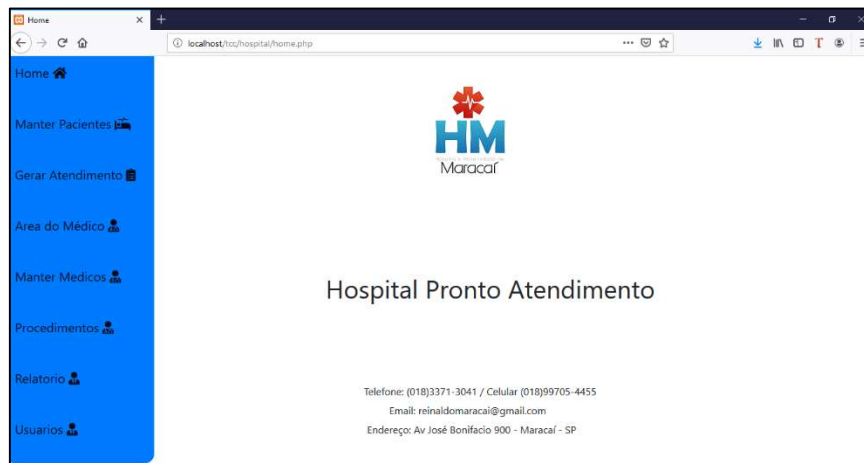


Figura 20 - Home Page

3.3.5. Manter Pacientes

A tela Manter Pacientes permite ao usuário realizar cadastros de novos pacientes, buscar pacientes cadastrados e verificar seus dados, buscar pacientes através do número do sus, conforme a figura 21.

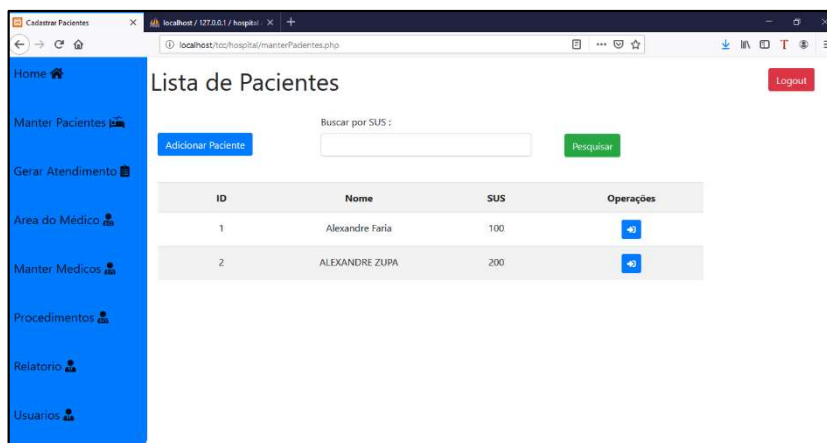


Figura 21 - Tela Manter Pacientes

3.3.6. atendimentos em Aberto

A tela de atendimentos em aberto permite ao médico acessar os atendimentos gerados, e realizar a execução do mesmo, conforme a figura 22.

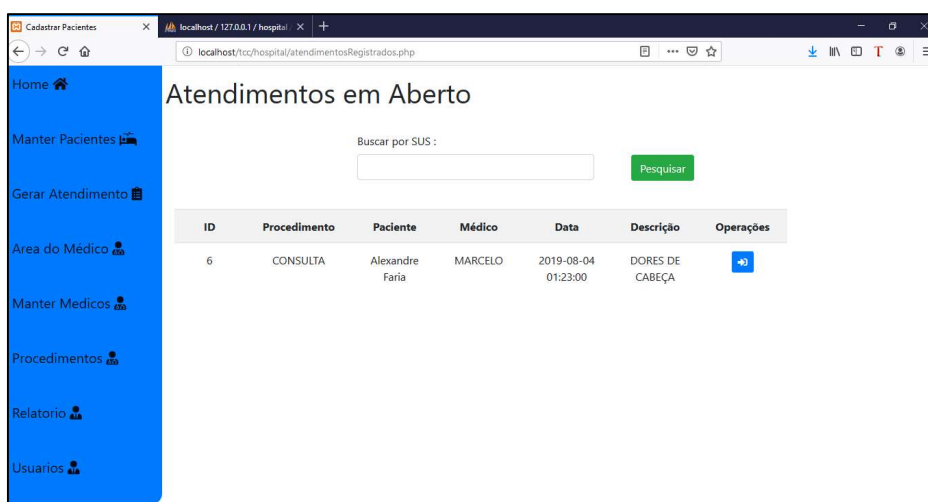


Figura 22 - Tela Atendimentos em aberto

3.3.7. Atendimentos Realizados

A tela de atendimentos em realizados permite ao usuário acessar os atendimentos que já foram realizados, em cada registro de atendimento na tabela possui um botão que permite gerar um relatório em PDF com todos os dados do atendimento, conforme a figura 23.

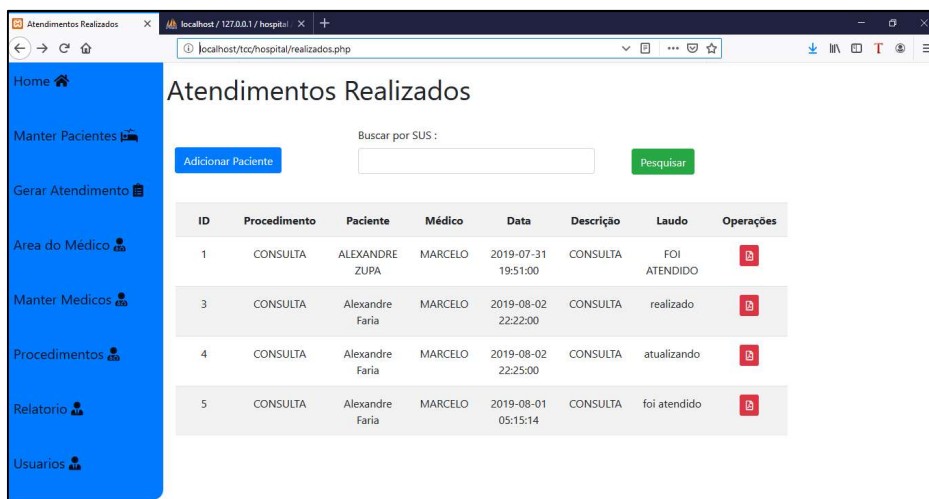


Figura 23 - Tela Atendimentos Realizados

3.3.8. Histórico do Paciente

A tela de histórico apresenta ao usuário, uma lista com todos os atendimentos que o paciente efetuou na entidade, podendo acessar um atendimento em específico e verificar as informações do mesmo, conforme a figura 24.

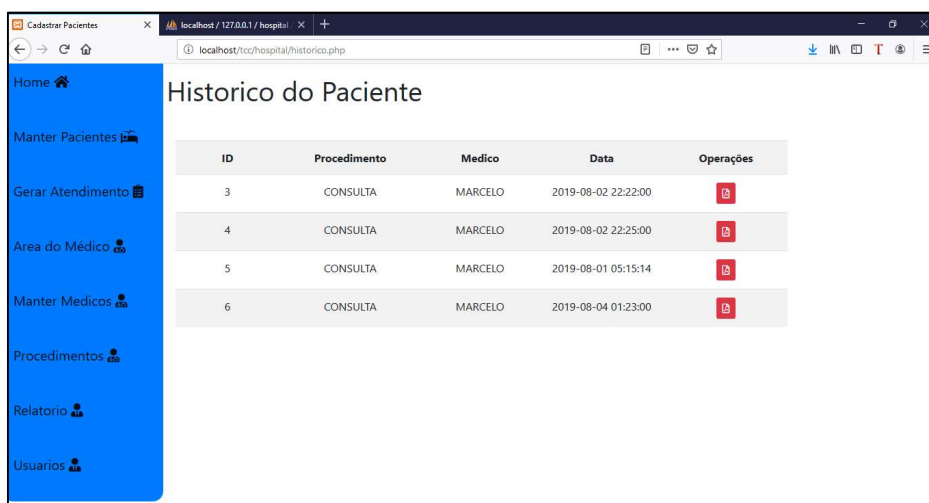


Figura 24 - Tela de Histórico do Paciente

3.3.9. Manter Médicos

A tela manter médicos apresenta ao usuário, uma lista com todos os médicos cadastrados na base de dados, possui um botão para cadastrar um novo medico, e em cada registro da lista, possui um botão para acessar os dados do médico, e altera-los, conforme a figura 25.

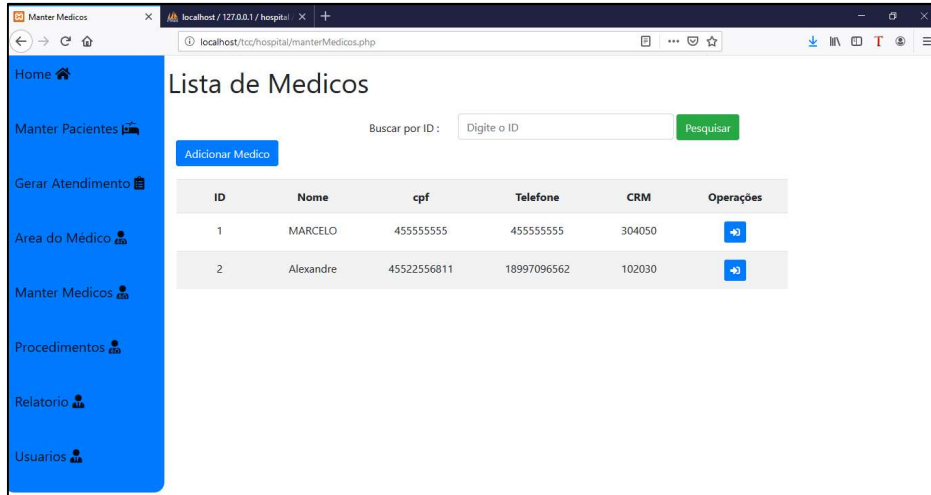


Figura 25 - Tela Manter Médicos

3.3.10. Cadastrar Usuário

A tela de cadastro de novo usuário apresenta ao usuário a possibilidade de incluir os usuários desejados ao acesso do sistema, o formulário possui validação de campos para que as informações solicitadas sejam preenchidas, caso contrário a ação não é executada, conforme a figura 26.

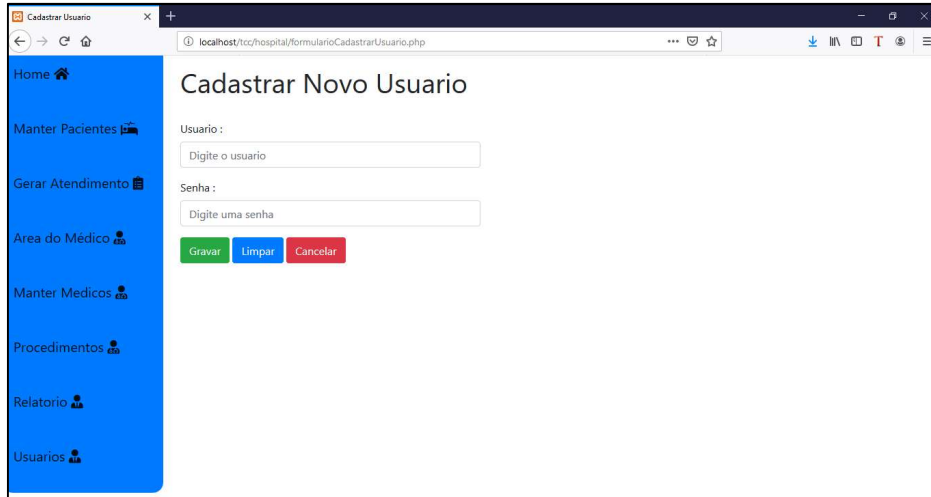


Figura 26 - Tela de Cadastrar Usuário

4. CONCLUSÃO

Para o desenvolvimento deste trabalho, inicialmente foi necessário realizar uma pesquisa na entidade escolhida, em busca de filtrar informações para gerar o planejamento projeto. Essa pesquisa foi importante para entendimento de como são os processos que estão inseridos em um pronto atendimento, como são as etapas e as informações relevantes para gerar um atendimento completo.

Após a modelagem do projeto, a segunda etapa foi o desenvolvimento do sistema, estudos sobre as tecnologias utilizadas e suas funcionalidades, onde foram adquiridos conhecimentos sobre um conjunto de ferramentas para o desenvolvimento de um software web.

Após o término do desenvolvimento do sistema, foram efetuados testes, com o objetivo de captar melhorias, novas funcionalidades e ajustes que proporcionassem uma experiência mais agradável e eficaz os usuários.

Conclui-se que o projeto desenvolvido, atendeu perfeitamente a necessidade que a entidade buscava para ter um sistema de atendimentos aos pacientes, de forma organizada, ágil e com dados íntegros.

5. REFERÊNCIAS

Barra DCC. Nascimento ERP, Martins JJ, Albuquerque GL, Erdmann AL. **Evolução histórica e impacto da tecnologia na área da saúde e da enfermagem**. Rev. Eletr. Enf. Disponível em: http://www.fen.ufg.br/revista/revista8_3/v8n3a13.htm Acesso em: 31/07/2019

AGRELA, Lucas. **Estas são as 20 linguagens de programação mais usadas**. Revista Exame Online Publicado em 23/04/2018. Disponível em: <https://exame.abril.com.br/tecnologia/estas-sao-as-20-linguagens-de-programacao-mais-usadas/> Acesso em: 31/07/2019

FERREIRA, Carlos. **7 motivos para aprender PHP em 2018**. Especializa Blog Publicado em 16/04/2018. Disponível em: <https://blog.especializati.com.br/7-motivos-para-aprender-php-em-2018/> Acesso em: 31/07/2019

TARSTC, Autor. **Login Cloud**. Bootsnipp. Disponível em: <https://bootsnipp.com/snippets/P2Xv1> Acesso em: 31/07/2019

Christiano. **Criptografia MD5**. DevMedia Publicado em 2007. Disponível em: <https://www.devmedia.com.br/criptografia-md5/2944> Acesso em: 31/07/2019

VALE, Christiano Rodarte. **Criptografia MD5**. DevMedia. Disponível em: <http://www.devmedia.com.br/post-2944-Criptografia-MD5.html>. Acesso em: 31/07/2019.

JUSTO, Andreia Silva. **Gerenciamento de Cronograma: o que é, qual a importância, como fazer e dicas práticas!**. Publicado em 06/02/2019. Disponível em: <https://www.euax.com.br/2019/02/gerenciamento-de-cronograma/> Acesso em: 31/07/2019

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