

# Sistema de Gerenciamento de Prontuários Médicos

João Pedro Garcia <sup>1</sup>Gonçalves, Daiane Mastrangelo <sup>2</sup>Tomazeti

Campus Hortolândia - Instituto Federal de Educação, Ciência e Tecnologia de São Paulo (IFSP) - 13183-250 - Vila São Pedro - Hortolândia - São Paulo - Brasil.

<sup>1</sup>joao.garcia@aluno.ifsp.edu.br, <sup>2</sup>daiianetomazeti@ifsp.edu.br

**Abstract.** *The Sistema de Gerenciamento de Prontuários Médicos project emerged within the scope of the IFSP to create a system that manages medical records, with the aim of facilitating the exchange of information between health professionals from the same clinic and about the same patient. To demonstrate to users that this system is different from others, we carried out a requirements survey comparing projects that already exist today. The objective of this project was to build a web system and the Waterfall methodology was used for its development, in addition to the requirements elicitation process, such as: use case diagram and user stories.*

**Resumo.** *O projeto Sistema de Gerenciamento de Prontuários Médicos surgiu no âmbito do IFSP para a criação de um sistema que gerencia prontuários médicos, com o intuito de facilitar o intercâmbio de informações entre os profissionais da saúde de uma mesma clínica e sobre um mesmo paciente. Para demonstrar aos usuários que este sistema é diferente dos demais, fizemos um levantamento de requisitos comparando projetos já existentes nos dias de hoje. O objetivo deste projeto foi a construção de um sistema web e para o seu desenvolvimento foi utilizada a metodologia Cascata, além do processo de elicitação de requisitos, tais como: diagrama de casos de uso e histórias de usuários.*

## 1. Introdução

A motivação deste trabalho surgiu ao observar ao longo do tempo o modo como são armazenadas as informações em clínicas médicas.

Por meio de observações feitas em diversas clínicas frequentadas por um dos autores deste trabalho, bem como sondagem informal com alguns profissionais da saúde, foi constatado que as informações sobre os pacientes são na maior parte das vezes armazenadas em folhas de papel. Quando se trata de uma clínica multidisciplinar, onde trabalham em conjunto vários profissionais da saúde acerca de um mesmo paciente, há necessidade de intercâmbio de informações entre eles, e armazenar dados em folhas de papel pode ser uma desvantagem em detrimento de um sistema que pode agilizar e otimizar o trabalho de todos.

Explorando estes cenários, o presente trabalho apresenta o desenvolvimento de uma aplicação *web* para instituições voltadas à área da saúde, podendo facilitar para estas o gerenciamento de horários e prontuários, sendo possível criar um banco de dados para gerenciar estas informações.

Uma vez determinado o objetivo deste trabalho, a implementação de um sistema *web* utilizando a linguagem de programação PHP, buscando agilidade e produtividade, foi

escolhida a ferramenta de apoio para o desenvolvimento o *Visual Studio Code*, uma *IDE* (um programa de computador que reúne características e ferramentas de apoio ao desenvolvimento de software com o objetivo de agilizar este processo) com suporte para todas as linguagens utilizadas, além de possuir uma boa comunidade e uma rica documentação.

O projeto foi organizado utilizando a metodologia cascata. Não se sabe ao certo quem criou este modelo, porém, um dos primeiros registros desta metodologia ocorreu em 1970 por Winston Walker Royce, onde Royce comenta sobre a experiência dele em gerenciamento de software e apresenta sugestões de melhorias para as formas de trabalho presenciadas ao longo da sua carreira.(PEREIRA 2017)

## **2. Justificativa**

Nos dias de hoje existem aplicações que desempenham algumas funções realizadas no projeto. Tendo isso em mente criou-se esta aplicação, que tem como objetivo desenvolver uma aplicação de gerenciamento de prontuários médicos, com o intuito de facilitar o intercâmbio de informações entre os profissionais da saúde de uma mesma clínica e sobre um mesmo paciente. O projeto será realizado utilizando a linguagem PHP, além de utilizar CSS, HTML e JavaScript.

## **3. Objetivos**

Os objetivos são responsáveis em explicar ao leitor os objetivos gerais e específicos do projeto.

### **3.1 Objetivos Gerais**

O objetivo geral deste projeto é o desenvolvimento de um *software web*. Segundo TotLab (2024) *web Software* é um *software* que você usa pela Internet com um navegador *web*. Você não precisa instalar nenhum CD, fazer *download* ou mesmo se preocupar com atualizações. O intuito do projeto é a criação de um sistema de gerenciamento de clínicas gratuito.

### **3.2 Objetivos Específicos**

Como o objetivo principal deste projeto é o desenvolvimento de um *software* que irá realizar o controle de um sistema com o intuito de armazenar dados de prontuários médicos de pacientes, a aplicação desenvolvida possui como diferencial o intercâmbio de informações entre os profissionais da saúde que atendem um mesmo paciente, bem como utilizar todo o programa de forma gratuita, em relação às atuais aplicações existentes.

## **4. Trabalhos Correlatos**

Trabalhos correlatos são projetos com propostas similares em relação ao projeto deste artigo, utilizados para a realização e desenvolvimento de ideias para este trabalho através de análise, teste ou pesquisa sobre os mesmos. Para este trabalho, foi pesquisado na internet sobre “sistemas de gerenciamento de clínicas” e dentre os listados, escolhidos três (3) que mais se aproximavam do descrito por duas profissionais que atuam em clínica multidisciplinar e que foram denominadas *stackholders* durante o desenvolvimento deste sistema.

### **4.1. Conclínica & IClinic**

Conclínica (2022) é um *software* para gestão de clínicas. Este sistema provê módulos como de controle financeiro, clínico, administrativo e de recepção. Dentre seus diferenciais é observável o suporte para telemedicina (com suporte para agendamento e realização de consultas *online*) e geração de relatórios.

### **4.2. IClinic**

IClinic (2022) é um sistema voltado para gestão clínica e de prontuário eletrônico. Este sistema possui funcionalidades como controle de agenda, prontuários, gestão financeira, controle de agendamentos, geração de relatórios e telemedicina. Dentre seus diferenciais, este possui assistente de utilização, onde, por etapas, os seus novos usuários podem aprender sobre o mesmo de forma sequencial.

### **4.3. Desenvolvimento de um sistema para atendimento à distância de pacientes com doença renal crônica pré-dialítica**

O sistema foi desenvolvido utilizando a linguagem Java, banco de dados MySQL e *framework* PrimeFaces e disponibilizado em um servidor de aplicações Glassfish.

J Bras Nefrol (2015) concluiu que esta ferramenta possibilitará aumentar a área de abrangência dos nefrologistas, diminuir os custos e aproximar o paciente ao seu médico da atenção básica, utilizando o Programa de Saúde de Família como interface entre o paciente e a atenção secundária nefrológica.





















### **4.4. Comparativo entre trabalhos**

Para a melhor compreensão das semelhanças e diferenças entre os trabalhos correlatos e o que foi desenvolvido neste trabalho, a Tabela 1 apresenta um comparativo entre as funcionalidades oferecidas entre tais sistemas.

No comparativo entre o Conclínica, o IClinic e o sistema desenvolvido apresenta o diferencial de permitir que todos os profissionais da saúde que atendem um mesmo paciente tenha acesso a todos os pareceres clínicos (intercâmbio de informações), permitindo assim uma visão mais geral e mais detalhada de cada caso, além de ser gratuito.

Em relação ao Sistema para atendimento à distância de pacientes com doença renal crônica pré-dialítica, o sistema aqui descrito oferece controle clínico e administrativo, o intercâmbio de informações e a gratuidade.

**Tabela 1. Comparativo entre trabalhos correlatos e o sistema deste presente trabalho.**

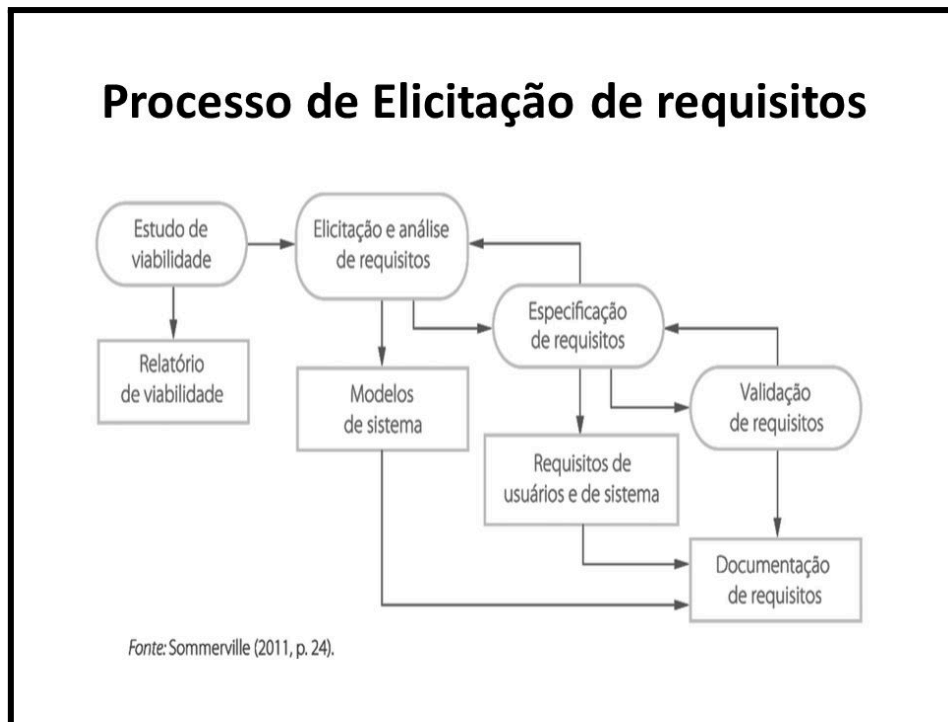
Sistemas	Controle Clínico	Controle Administrativo	Suporte para telemedicina	Gratuito	Intercâmbio de informações
Sistema de Gerenciamento de Prontuários Médicos de Pacientes			Por meio de trabalhos futuros 		
Conclínica					
IClinic					
Desenvolvimento de um sistema para atendimento à distância de pacientes com doença renal crônica pré-dialítica					

## 5. Fundamentação Teórica

Esta seção apresentará os fundamentos da implementação do sistema descrito neste artigo, de seu modelo de desenvolvimento utilizado, de seus conceitos arquiteturais, da elicitação de requisitos e parte das ferramentas e tecnologias utilizadas no decorrer do processo de desenvolvimento.

### 5.1. Elicitação de requisitos

Segundo Sommerville (2011), utilizar engenharia de requisitos como o processo de descobrir, analisar, documentar e verificar as funções e restrições do sistema tem como objetivo criar e manter um documento de requisitos de sistema. A Figura 1 mostra o passo a passo do processo de elicitação de requisitos.



**Figura 1. Exemplo da elicitação de requisitos Sommerville(2011, p. 24).**

### 5.1.1. Estudo de viabilidade

Segundo Pressman (2006, p. 118), o analista, juntamente com o cliente, tenta identificar qual o tamanho e a funcionalidade do sistema, realizando um estudo de viabilidade, por fim obtendo um documento denominado de escopo do projeto.

### 5.1.2. Levantamento e análise de requisitos

De acordo com Sommerville (2007, p. 98), a atividade de análise de requisitos visa priorizar e resolver conflitos entre requisitos, pois quando vários usuários participam desse processo, é inevitável que ocorra contradição entre requisitos levantados de usuários distintos.

### 5.1.3. Documentação dos requisitos

Segundo Pressman (2006, p. 120), uma especificação pode ser um documento escrito, um modelo gráfico, ou matemático formal, um protótipo ou qualquer combinação desses documentos.

### 5.1.4. Validação dos requisitos

Sommerville (2007, p. 105) aponta como objetivo garantir que a necessidade real do usuário esteja descrita corretamente no documento de especificação dos requisitos. A validação é extremamente importante, pois o custo para correção de um requisito nessa fase é bem inferior ao custo nas fases posteriores, como implementação ou testes. Isso ocorre, pois se defeitos encontrados nessas fases, os requisitos devem ser novamente levantados e posteriormente são implementados e testados.

Pressman (2006, p. 120) afirma que os requisitos devem ser examinados para que sejam encontradas inconsistências, ambiguidades e omissões. Para tal atividade, Sommerville

(2007, p. 106) propõe algumas técnicas de validação, tais como revisões de requisitos, prototipação e geração de casos de teste.

### 5.1.5. Considerações finais

Por fim, a atividade de validação dos requisitos garante que as informações contidas no documento de requisitos realmente atendem às necessidades do usuário. Aprovado, a documentação foi referência para o restante da execução do projeto de desenvolvimento do *software*.

## 5.2. Padrão arquitetural MVC

MVC é o acrônimo de *Model-View-Controller*. Pressman (2006) relata que o MVC é um padrão de projeto de *software*, ou padrão de arquitetura de *software* formulado na década de 1970, focado no reuso de código e a separação de conceitos em três camadas. A Figura 2 mostra um exemplo de funcionamento da arquitetura estrutural MVC.

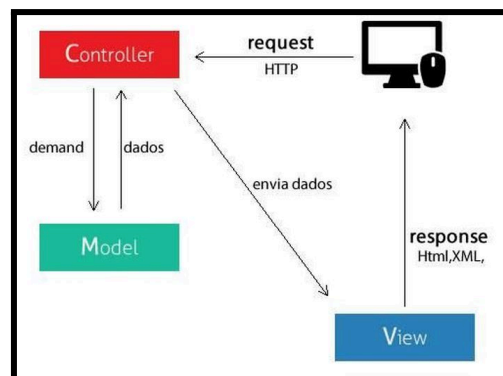


Figura 2. Exemplo do Padrão arquitetural MVC.

## 5.3. Banco de dados

Segundo Oracle (2022), um banco de dados é uma coleção organizada de informações - ou dados - estruturadas, normalmente armazenadas eletronicamente em um sistema de computador. Um banco de dados é geralmente controlado por um sistema de gerenciamento de banco de dados (SGBD). Juntos, os dados e o SGBD, juntamente com os aplicativos associados a eles são chamados de sistema de banco de dados, geralmente abreviados para apenas banco de dados.

Um diagrama entidade relacionamento (ER) é um tipo de fluxograma que ilustra como “entidades”, pessoas, objetos ou conceitos, se relacionam entre si dentro de um sistema (Nogueira, 1988). Diagramas ER são mais utilizados para projetar ou depurar bancos de dados relacionais nas áreas de Engenharia de *Software*, sistemas de informações empresariais, educação e pesquisa (Genong et al., 2010; Kawabata, 2015).

A Figura 3 apresenta o Diagrama de entidade e relacionamento do projeto, e na Figura 4 o exemplo de funcionamento de um banco de dados e como ele é utilizado nos sistemas.

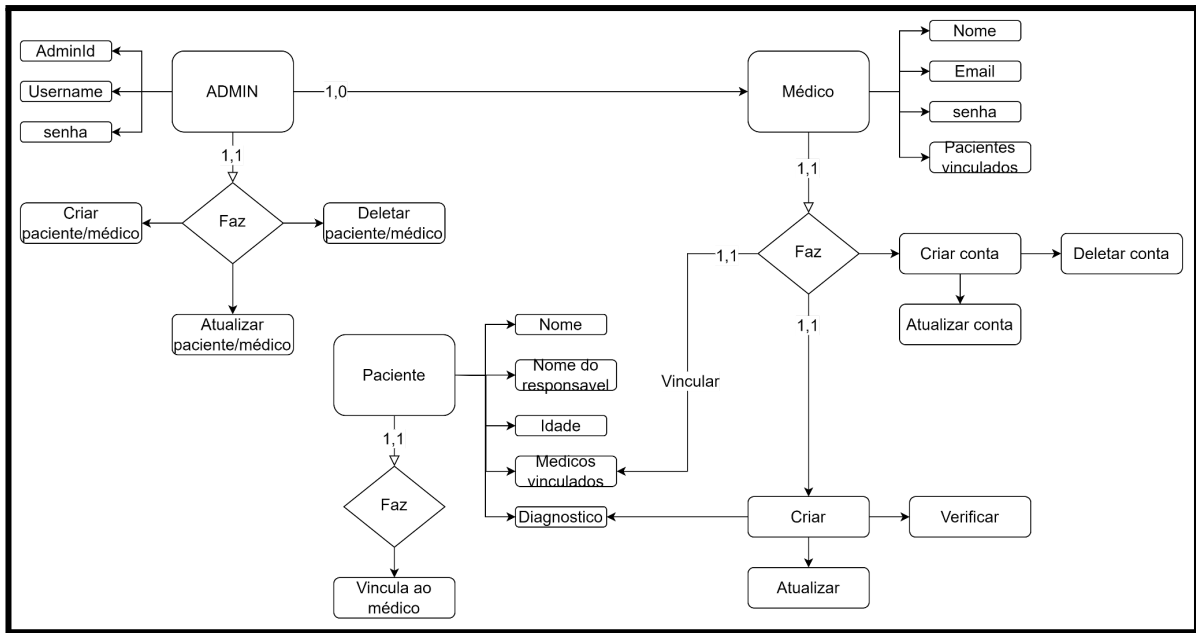


Figura 3. Diagrama de entidade e relacionamento do projeto.

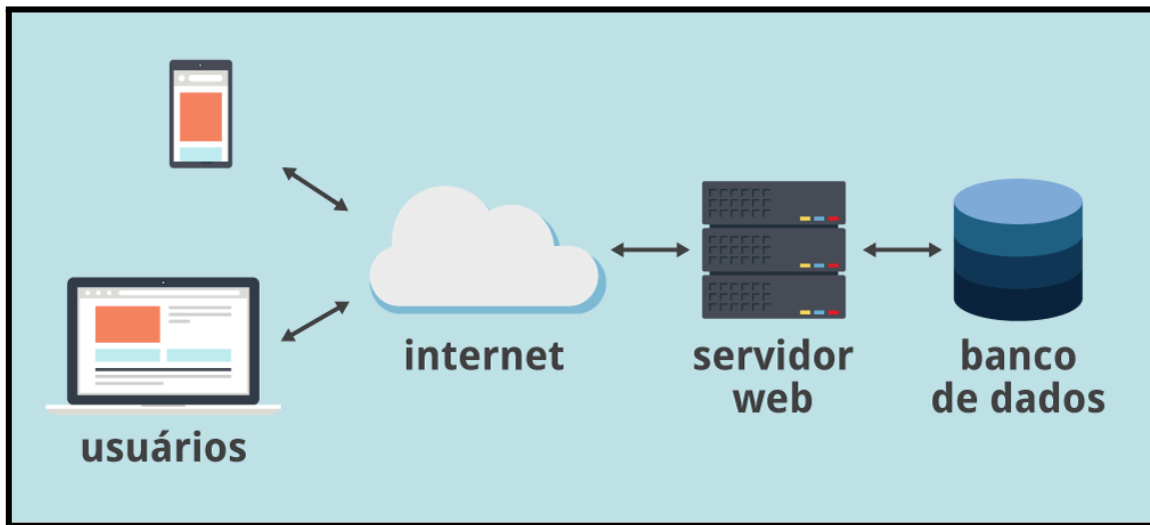


Figura 4. Exemplo do funcionamento do banco de dados.

## 5.4. PHP

PHP é uma linguagem interpretada livre, usada originalmente apenas para o desenvolvimento de aplicações presentes e atuantes no lado do servidor, capazes de gerar conteúdo dinâmico na *World Wide Web*. (RASMUS, 1944)

## 5.5. CSS

O *Cascading Style Sheets* (CSS) é uma "folha de estilo" composta por "camadas" e utilizada para definir a apresentação (aparência) em páginas da internet que adotam para o seu desenvolvimento linguagens de marcação (como XML, HTML e XHTML). (HÅKON, 1944).

## 5.6. HTML

*HyperText Markup Language* (HTML) é o componente base da *web*. Isso quer dizer que ele permite a construção de *websites* e a inserção de novos conteúdos, como imagens e vídeos, por meio dos hipertextos. (TIM BORNER-LEE, 1989).

## 5.7. JavaScript

*JavaScript* é uma linguagem de programação que permite a você implementar itens dinâmicos em páginas *web*. Toda vez que uma página da *web* faz mais do que simplesmente mostrar a você informação estática, mostrando conteúdo que se atualiza em um intervalo de tempo, mapas interativos ou gráficos 2D/3D animados, etc. (BRENDAN, 1955).

## 5.8. SQL

SQL significa *Standard Query Language*, linguagem padrão para realizar queries. A linguagem SQL é utilizada de maneira relativamente parecida entre os principais bancos de dados relacionais do mercado. (IBM, 1970).

## 5.9. MySQL

O MySQL é um sistema de gerenciamento de banco de dados relacional de código aberto (RDBMS) suportado pela *Oracle* e baseado em linguagem de consulta estruturada (SQL). O sistema funciona em praticamente todas as plataformas, incluindo *Linux*, *UNIX* e *Windows*. (MONTY, 1955).

## 5.10. PHPMyAdmin

*PHPMyAdmin* é uma ferramenta de *software* livre escrita em PHP que se destina a lidar com a administração de um servidor de banco de dados MySQL ou MariaDB. (OLIVER, 2001).

## 5.11. Visual Studio Code

O *Visual Studio Code* (VS Code) é um editor de código aberto desenvolvido pela *Microsoft*. A saber, ele está disponível para *Windows*, *Mac* e *Linux*. É criado com *Electron*, ferramenta criada pelo *GitHub* que permite a criação de *softwares Desktop* com HTML, CSS e *JavaScript*. (MICROSOFT, 2015).

## 5.12. Xampp

*Xampp* é um pacote de servidores de código aberto que provê a instalação do *Apache*(com servidor HTTP), assim como contém a instalação do banco de dados *MYSQL* e a instalação do *PHP*. (STEFFEN, 2002).



## 6. Metodologia

Durante o desenvolvimento do sistema deste projeto do Trabalho de Conclusão de Curso (TCC), houve apoio de dois *stakeholders* sendo uma terapeuta ocupacional e uma psicopedagoga que trabalham em clínica multidisciplinar, com o objetivo de fornecer informações para iniciar o desenvolvimento, utilizando o sistema cascata.

Também conhecido como Modelo Sequencial Linear, o nome cascata vem devido à sua forma sequencial cascadeada que acontece de uma fase para outra. Cada pedaço (cascata) é desenvolvido de forma linear, e em seguida exposto aos comentários dos clientes. (PRESSMAN 2006)

## 7. Desenvolvimento

Esta sessão retrata a implementação do sistema, a definição de sua arquitetura, seu levantamento de requisitos e seu modelo de dados.

### 7.1. Estudo de viabilidade

Após observação da quantidade de *softwares* disponíveis e receber informações das profissionais de apoio (*stakeholders*) sobre suas dificuldades em obter informações de maneira sistematizada e rápida sobre seus pacientes em comum, além de como as aplicações existentes estão longe do orçamento disponível dos mesmos, foi perceptível que existia a possibilidade para o desenvolvimento deste *software*. Assim sendo, após reunião com as profissionais, deu-se início ao processo de levantamento e análise de requisitos.

### 7.2. Levantamento, Análise e Documentação de Requisitos

Sommerville (2018) afirma que a Análise de Requisitos é a área de conhecimento que descreve as tarefas e técnicas utilizadas por um analista de negócios para analisar requisitos declarados, no intuito de definir as capacidades requeridas de uma solução potencial para atender às necessidades das partes interessadas.

Requisitos funcionais e não funcionais do projeto:

Atenta-se que onde se lê “médico” pode ser qualquer profissional da saúde que atenda o paciente, levando em consideração o contexto de clínica multidisciplinar.

- Requisitos funcionais:
  1. Criação de diversos tipos de perfis;
  2. Geração de relatório periodicamente;
  3. Diferentes níveis de acesso com base no tipo de perfil criado.
  4. Registrar o histórico do paciente.
  5. Acompanhar o desenvolvimento do paciente.
  6. O paciente pode possuir mais de um médico.
  
- Requisitos não funcionais:
  1. Conexão ao banco de dados local MySQL;

2. Responsividade com dispositivos *mobile*;
3. Desenvolvido localmente para uma maior segurança com os dados médicos fictícios;
4. Padrão arquitetural MVC;

O diagrama de caso de uso descreve a funcionalidade proposta para um novo sistema que será projetado (Sommerville, 2018). A Figura 5 mostra o Diagrama de casos de uso do projeto.

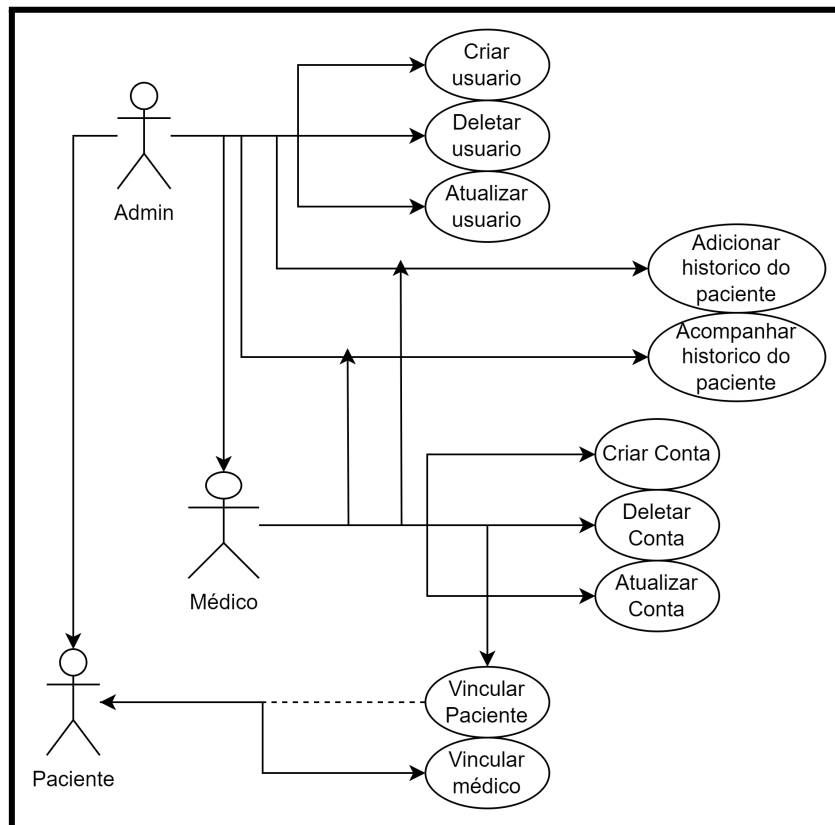


Figura 5. Diagrama de casos de uso.

### 7.3. Elaboração de histórias de usuário para o portal

As histórias de usuário foram levantadas em uma reunião com as profissionais da saúde (*stackholders*).

1	Como admin, eu quero ter o poder de adicionar todos os demais usuários;
2	Como admin, eu quero ter o poder de deletar todos os demais usuários;
3	Como admin, eu quero ter o poder de atualizar todos os demais usuários;

4	Como médico, eu quero ter o poder de verificar os históricos do paciente;
5	Como médico, eu quero ter o poder de criar minha conta;
6	Como médico, eu quero ter o poder de atualizar meus dados;
7	Como médico, eu quero ter o poder de deletar minha conta;
8	Como médico, eu quero ter o poder de me vincular com um paciente;
9	Como médico, eu quero ter o poder de escrever o histórico do paciente;
10	Como médico, eu quero ter o poder de atualizar o histórico do paciente;
11	Como paciente, eu quero ter múltiplos médicos.

Com o levantamento das histórias de usuários também foi obtido o conhecimento dos diferentes tipos de usuários que irão compor o sistema. Foram observados três níveis de usuários, sendo eles: Médico (profissionais da saúde que atendem o mesmo paciente), Admin, Paciente. O papel do Admin é de vincular um Paciente a um Médico e também de adicionar um paciente. O papel de um Médico é de visualizar todos os históricos do Paciente, adicionar e atualizar um histórico.

#### 7.4. Protótipo visual do portal

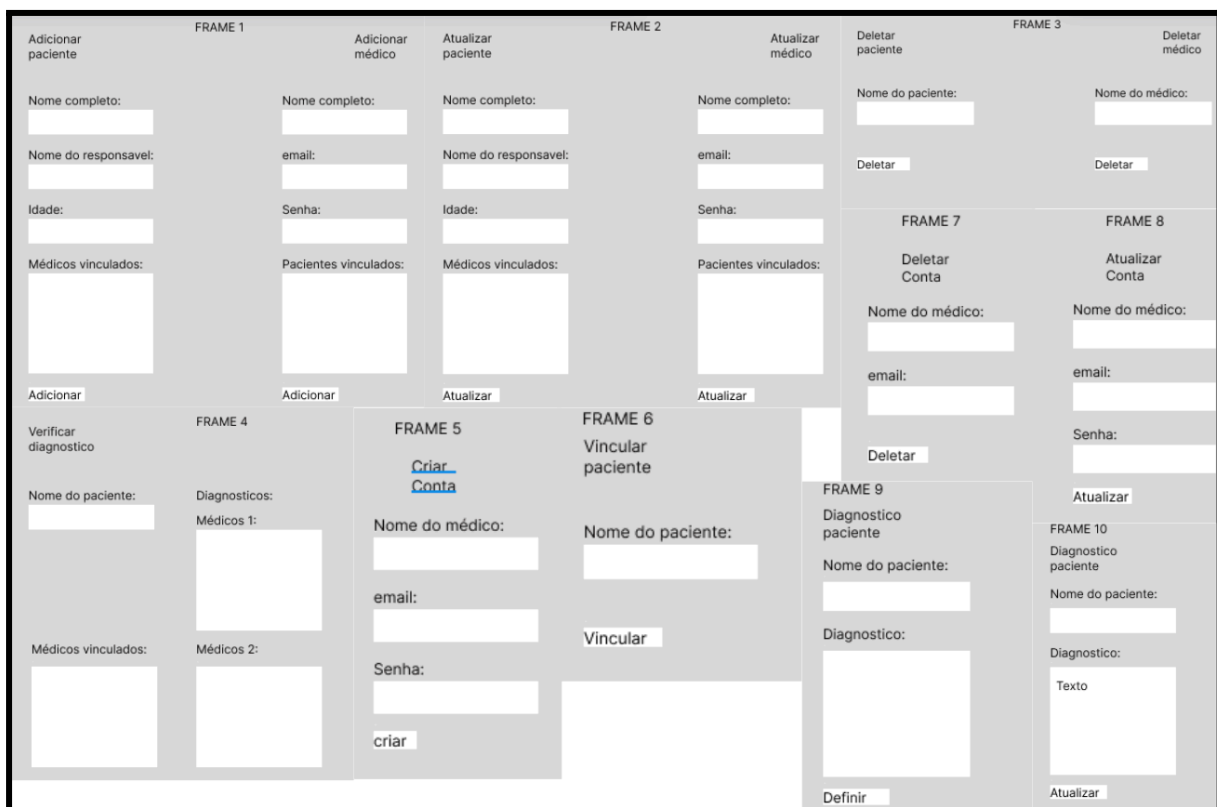
Para criar os protótipos de *design* deste projeto usamos *wireframes*. *Wireframe* pode ser definido como um esqueleto ou uma versão bastante primitiva do visual final de um projeto. Segundo Becker (2019), *wireframes* são como desenhos de rascunhos do *design* do sistema. Para criação dos mesmos foi usada a plataforma Miro. Com esta plataforma é possível desenhar elementos gráficos de sistemas com a opção de trabalho simultâneo e remoto.

A parte criativa do *design* foi elaborada analisando as histórias de usuário e inserindo as suas necessidades dentro das diferentes telas do sistema. Com isso cada tela correspondeu às histórias da seguinte forma:

- Frame 1: História 1,11;
- Frame 2: História 3;
- Frame 3: História 2;
- Frame 4: História 4;
- Frame 5: História 5;
- Frame 6: História 8;

- Frame 7: História 7;
- Frame 8: História 6;
- Frame 9: História 9;
- Frame 10: História 10.

Na Figura 6 demonstramos os *wireframes* do projeto. Após esta etapa foi iniciado o trabalho de codificação do *site*, passando o protótipo (Figura 6) representado por um desenho para um *site* real utilizando a linguagem de marcação HTML5 e CSS3. Esta primeira versão do *site* foi criada em um servidor Apache em um computador local funcionando somente com acesso dentro da mesma rede de conexão com a internet.



**Figura 6. Wireframes contendo a primeira versão do design do sistema.**

### 7.4.1 Design final do portal

Com o *design* das páginas pronto (Figura 7,8), foi adicionada uma base de dados para trabalhar com dados dentro das páginas, possibilitando utilizar os dados salvos na base de dados como, por exemplo, uma lista com todos os pacientes existentes.

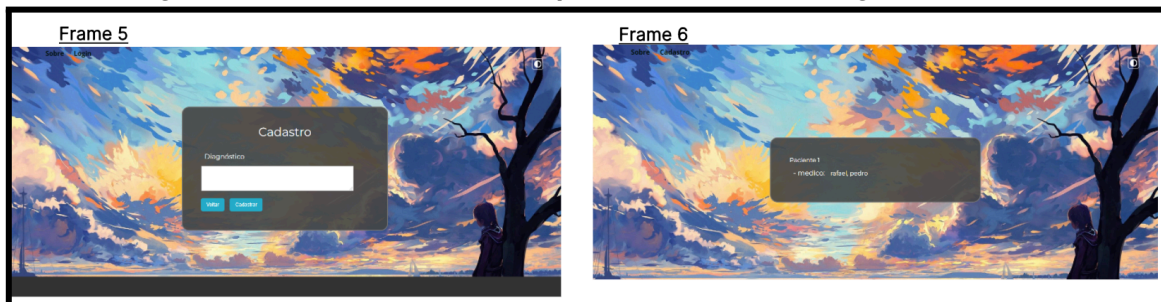
As páginas também passaram a ter códigos de programação. Para isso foi utilizado a linguagem PHP. Foi criada uma conexão com o banco de dados local (Figura 9) para ser possível adicionar as funcionalidades do programa, tais como: adicionar, atualizar, deletar médicos e pacientes (Figura 10).

- Frame 1: História 1,2,3;
- Frame 2: História 4;

- Frame 3: História 5,8;
- Frame 4: História 6,7;
- Frame 5: História 9,10;
- Frame 6: História 11.



**Figura 7. Wireframes contendo a primeira versão do design do sistema.**



**Figura 8. Wireframes contendo a primeira versão do design do sistema.**

```

conexao.php
1  <?php
2  $servidor = "localhost";
3  $usuario = "root";
4  $senha = "";
5  $dbname = "id18600975_localhost";
6
7  //Criar a conexao
8  $conn = mysqli_connect($servidor, $usuario, $senha, $dbname);

```

**Figura 9. Exemplo de conexão com o banco de dados.**

```

FuncãoLogin.php
1 <?php
2 session_start();
3 include_once 'conexao.php';
4
5 if(empty($_POST['nome']) || empty($_POST['senha'])) {
6     header('Location: Login.php');
7     exit();
8 }
9
10 $nome = $_POST['nome'];
11 $senha = MD5($_POST['senha']);
12 $sql = "SELECT * FROM usuario WHERE nome = '$nome' AND senha = '$senha'";
13
14 if ($result = mysqli_query($conn,$sql)) {
15     if (mysqli_num_rows($result) >= 1) {
16         $_SESSION['nome'] = $nome;
17         header("Location: ./index2.php");
18         exit();
19     } else {
20         $_SESSION['nao_autenticado'] = true;
21         header('Location: ./Login.php');
22         exit();
23     }
24 }

```

Figura 10. Exemplo de função em PHP.

## 7.5 Testes Unitários

De acordo com Sommerville (2018), os testes pretendem mostrar que um sistema faz o que foi orientado a fazer e descobrir os erros antes que ele seja colocado em desenvolvimento. Com isso, testes unitários foram implementados dentro da API do sistema, a fim de validar entradas e saídas e garantir o funcionamento adequado da solução. Para a construção das mesmas, o *framework* PHPUnit, do qual seu principal objetivo é executar um trecho de código de forma automática e exibir o resultado de todos os testes [ESTEVIÃO 2020].

Um trecho de código dos testes unitários implementados pode ser visualizado na Figura 11. Nela, observa-se um método de teste para validar se o *login* do usuário foi realizado corretamente.

```

public function testLoginSuccess()
{
    protected $usuario;
    protected $senha;
    protected $conexao;

    public function setUp()
    {
        $this->usuario = new usuario("José", "12345678");
        $this->conexao = new Connection('localhost', 'root', 'dfre43e$$');
    }

    public function tearDown()
    {
        $this->conexao->close();
    }
}

```

Figura 11. Exemplo de teste unitário em PHP.

## 7.6 Testes de funcionalidade

Além dos testes unitários, foram realizados testes de funcionalidade durante o processo de desenvolvimento. Foram utilizadas as histórias de usuário como base para realizar o teste dos fluxos das funcionalidades do sistema, sendo assim possível descobrir bugs e corrigi-los.

O teste de funcionalidade tem como função avaliar as funções do sistema observando se estão funcionando corretamente, além de checar se o mesmo está adequado aos objetivos do negócio [CARVALHO 2019].

## 8. Conclusão

O trabalho apresentado desenvolveu um sistema *web* para a realização de um programa de gerenciamento de prontuários médicos. Quando disponibilizada esta ferramenta poderá auxiliar o gerenciamento de várias clínicas multidisciplinares, pois trata-se de um sistema gratuito e que facilita o intercâmbio de informações entre os profissionais da saúde de uma mesma clínica sobre um mesmo paciente. Desta maneira pode-se dizer que esta ferramenta poderá otimizar estudos de casos específicos, além de contribuir no quesito financeiro.

Como trabalhos futuros, pode-se citar: um meio de migrar o sistema que atualmente é executado localmente para um sistema *cloud*.

Com o desenvolvimento deste trabalho, foi possível aplicar alguns conhecimentos obtidos em várias disciplinas no decorrer do Curso Superior de Análise e Desenvolvimento de Sistemas. Dentre elas, estão: Arquitetura de Software; Análise Orientada a Objetos; Banco de Dados I e II; Desenvolvimento Web; Engenharia de Software; Interação Humano-Computador; Metodologia de Pesquisa e Projeto de Sistemas para planejamento e desenvolvimento teórico-técnico.

## Referências Bibliográficas

[BECKER 2019] BECKER,L.(2019) “Wireframes, o que são e por que os utilizamos?” Disponível em:

<<https://www.organicadigital.com/blog/o-que-sao-wireframes-e-por-que-os-utilizamos/>>.

Acesso em 20 de agosto de 2022.

[CARVALHO 2019] CARVALHO, I. (2019). Teste de funcionalidade, teste de desempenho e teste de usabilidade (part 1). Disponível em: <https://tinyurl.com/2v5hn9nb>. Acesso em 24 de novembro de 2023.

[CONCLINICA 2022] CONCLINICA. (2022) Disponível em:

[https://conclinica.com.br/?gclid=EAIaIQobChMIvo6qk\\_KP-AIVYG1vBB24gwL2EAAYASAAEgLnOvD\\_BwE](https://conclinica.com.br/?gclid=EAIaIQobChMIvo6qk_KP-AIVYG1vBB24gwL2EAAYASAAEgLnOvD_BwE). Acesso em 20 de agosto de 2022.

[CSS 2022] CSS (2022) Disponível em:

<https://www.tecmundo.com.br/programacao/2705-o-que-e-css-.htm>. Acesso em 20 de agosto de 2022.

[ESTEVÃO 2020] ESTEVÃO. (2020) Disponível em:

<https://www.devmedia.com.br/teste-unitario-com-phpunit/41231>. Acesso em 20 de janeiro de 2024.



Genong, Y., Liping, D., Zhang, B. & Wang, H. (2010). Coordination Through Geospatial Web Service Workflow in the Sensor Web Environment. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing , 3(4): 433–441.

[HTML 2022] HTML (2022). Disponível em:  
<https://developer.mozilla.org/pt-BR/docs/Web/HTML>. Acesso em 20 de agosto de 2022.

[ICLINIC 2022] ICLINIC. (2022). Disponível em:<https://iclinic.com.br/>. Acesso em 20 de agosto de 2022.

Jana Pereira (2017). A grande mentira do waterfall. Disponível em:  
<http://agile.pub/assuntos-diversos/a-grande-mentira-do-waterfall/>. Acesso em 12 de março de 2024.

[JAVASCRIPT 2022] JAVASCRIPT (2022). Disponível em:  
[https://developer.mozilla.org/pt-BR/docs/Learn/JavaScript/First\\_steps/What\\_is\\_JavaScript](https://developer.mozilla.org/pt-BR/docs/Learn/JavaScript/First_steps/What_is_JavaScript). Acesso em 20 de agosto de 2022.

[MYSQL 2022] MySQL (2022). Disponível em:  
<https://www.tecmundo.com.br/software/223924-mysql-usar-o-sistema.htm>. Acesso em 20 de agosto de 2022.

[MVC 2022] MVC (2022). Disponível em:  
<https://www.devmedia.com.br/introducao-ao-padrao-mvc/29308>. Acesso em 20 de agosto de 2022.

[NEFROL 2015] NEFROL.(2015) “Desenvolvimento de um sistema para atendimento à distância de pacientes com doença renal crônica pré-dialítica.” Disponível em:<https://www.scielo.br/j/jbn/a/YJNtcZXt9fbsNVGDD4t5BPG/?lang=pt> 28 de fevereiro de 2024.

Nogueira, D. L. (1988). Ferramentas automatizadas para apoio ao projeto estruturado: uma aplicação do diagrama de entidade-relacionamento. 336f. Tese (Doutorado em Ciências em Engenharia de Sistemas e Computação) – Universidade Federal do Rio de Janeiro, Rio de Janeiro, RJ.

[PHP 2022] PHP (2022). Disponível em:  
[https://www.php.net/manual/pt\\_BR/intro-what-is.php](https://www.php.net/manual/pt_BR/intro-what-is.php). Acesso em 20 de agosto de 2022.

[PHPMYADMIN 2022] PHPMYADMIN (2022). Disponível em:  
[https://docs.phpmyadmin.net/pt\\_BR/latest/intro.html](https://docs.phpmyadmin.net/pt_BR/latest/intro.html). Acesso em 20 de agosto de 2022.

[ROGER 2021] ROGER, P. (2021). Engenharia de software. 05 de julho de 2021, AMGH. Acesso em 27 de agosto de 2022.

[SOMMERVILLE 2018] SOMMERVILLE, I. (2018). Engenharia de Software. Pearson, 10th edition. Acesso em 27 de agosto de 2022.

[SQL 2022] SQL (2022). Disponível em:  
<https://www.devmedia.com.br/guia/guia-completo-de-sql/38314>. Acesso em 20 de agosto de 2022.



[VS CODE 2022] VS CODE. (2022). Disponível em: <https://code.visualstudio.com/>. Acesso em 20 de agosto de 2022.

# Documento Digitalizado Restrito

## Artigo de TCC do aluno João Pedro Garcia Gonçalves

**Assunto:** Artigo de TCC do aluno João Pedro Garcia Gonçalves

**Assinado por:** Daiane Tomazeti

**Tipo do Documento:** Comprovante

**Situação:** Finalizado

**Nível de Acesso:** Restrito

**Hipótese Legal:** Informação Pessoal - dados pessoais e dados pessoais sensíveis (Art. 31 da Lei nº 12.527/2011)

**Tipo do Conferência:** Cópia Simples

Documento assinado eletronicamente por:

- Daiane Mastrangelo Tomazeti, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 18/03/2024 10:39:06.

Este documento foi armazenado no SUAP em 18/03/2024. Para comprovar sua integridade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifsp.edu.br/verificar-documento-externo/> e forneça os dados abaixo:

**Código Verificador:** 1614910

**Código de Autenticação:** 822633022a

