# SISTEMA DE AGENDAMENTOS SIMPLIFICADO PARA CLÍNICAS DE PEQUENO PORTE

**Celso Guedes Gomes**[1]

[1]Instituto Federal de Educação, Ciência e Tecnologia de Rondônia - IFRO

sucelsos@gmail.com

***Resumo.*** *O sistema de saúde enfrentou grandes desafios após a pandemia do COVID-19, que exigiu medidas de prevenção e controle da contaminação. Nesse contexto, a inclusão digital e o uso massivo dos meios de tecnologia se tornaram essenciais para facilitar o acesso aos serviços de saúde, principalmente na área médica. Este trabalho tem como objetivo criar um sistema web para gerenciar agendamentos em clínicas de pequeno porte, solução tecnológica que visa melhorar a qualidade e a eficiência do atendimento médico. O sistema de agendamentos foi desenvolvido em linguagem PHP, com o framework CodeIgniter e Bootstrap, que são ferramentas que facilitam o desenvolvimento web e a criação de interfaces responsivas. Além disso, o sistema utiliza JavaScript para adicionar interatividade e dinamismo às páginas web, e banco de dados MySQL para armazenar e gerenciar as informações dos usuários e dos agendamentos. O produto tecnológico resultante deste trabalho é o sistema web chamado SINTEAG, que atende a demanda de agendamento de clínicas de pequeno porte.*

## 1. Introdução

Atualmente, destaca-se após a pandemia do COVID-19[1], muitos setores da sociedade, seja na iniciativa privada ou pública, buscam a combinação entre o uso da informação e a qualidade na prestação de serviços. A qualidade dos serviços passou a ser uma necessidade do mercado, e não apenas uma forma de diferenciar os prestadores de serviços. Conforme publicado em [MoneyLab, 2021], nos últimos meses o setor de tecnologia da informação comprovou sua importância em fundamentos essenciais, como inovação, segurança e transformação digital. Dentro dessa conjuntura, a área da saúde não poderia ficar atrás desses avanços.

Os riscos de contaminação por COVID-19 ainda são grandes. Apenas em Rondônia, de acordo com CONASS [2023], até 8 de maio de 2023, foram registrados 12.953 casos de contaminação por COVID-19, sendo que desses, 41 resultaram em óbito. Desde o início da pandemia, foram registrados um total de 484.626 casos de contaminação e 7.440 óbitos [BRASIL, 2023]. Além do COVID-19, existem outras doenças transmissíveis, e por isso, desde o início da pandemia, o Ministério da Saúde recomenda evitar locais com aglomeração de pessoas.

As Tecnologias da Informação e Comunicação (TIC) voltadas para o setor da saúde possuem inúmeras ferramentas para armazenar e processar dados, além de fornecer informações importantes em tempo real. Essas ferramentas facilitam a organização

[1]Infecção respiratória aguda causada pelo coronavírus SARS-CoV-2, potencialmente grave, de elevada transmissibilidade e de distribuição global.

desses dados e informações, tanto para os profissionais que as utilizam, quanto para os pacientes, que são melhor atendidos e têm acesso a informações de forma mais prática e rápida. Conforme apontado por Pinochet [2011], o gerenciamento de dados em setores hospitalares e áreas afins é um componente essencial do processo de assistência aos pacientes. Problemas com o gerenciamento de dados são ainda mais desafiadores devido ao crescimento exponencial na quantidade de informações a serem tratadas, bem como ao número de especialistas envolvidos no processo, que requerem acesso a essas informações em tempo real.

Em clínicas ou cooperativas de pequeno porte voltadas para áreas como fisioterapia, quiropraxia, massoterapia, entre outras, geralmente há apenas os profissionais específicos de cada área da saúde. Esses profissionais muitas vezes precisam atender e gerenciar horários, movimentação financeira e insumos por conta própria, o que pode gerar desperdício de tempo, aglomeração de pessoas e filas de espera. Portanto, é necessário otimizar essas atividades. Surge então a pergunta: como organizar o trabalho desses profissionais, evitando a concentração de pessoas e a possível disseminação de doenças, incluindo a COVID-19, por meio das Tecnologias da Informação e Comunicação?

As soluções de agendamento online para clínicas e consultórios médicos surgem como uma alternativa para melhorar a organização e qualidade do atendimento ao paciente, proporcionando facilidade de acesso e rapidez no agendamento, independentemente da especialidade médica Morais [2020]. Além de simplificar o contato direto com a clínica, aumentar a produtividade da equipe clínica e reduzir os custos com ligações, o agendamento online pode aumentar a presença virtual da clínica, entre outros benefícios.

Com a criação de um *software* que gerencie os atendimentos, permitindo agendamentos de horários e armazenamento de dados dos clientes, como diagnósticos, tratamentos, endereços e telefones de contato, o tempo do profissional pode ser otimizado, proporcionando mais tempo para exercer sua atividade e beneficiando aqueles que necessitam de seus serviços. Além disso, essa solução poderia tornar o atendimento mais eficiente e evitar um número maior de pessoas nas salas de espera.

Existem aplicativos com as funcionalidades descritas anteriormente, porém estes são ferramentas comerciais, de alto custo, e geralmente voltados para o atendimento de clínicas e hospitais de grande porte, como por exemplo: ClinicWeb[2], Gestaods[3], Iclinic[4], entre outros. Este trabalho propõe criar uma ferramenta simplificada, gratuita e de código aberto.

A proposta deste trabalho é criar uma ferramenta voltada mais especificamente para esse nicho de mercado: profissionais e prestadores de serviços que, mesmo em cooperativas, realizam atendimentos aos clientes como autônomos e gerenciam suas próprias necessidades, de acordo com os requisitos levantados junto aos profissionais da área.

## 2. Objetivos

Este trabalho teve como objetivo criar um sistema web gratuito e de código aberto para gerenciar agendamentos, com controle e armazenamento de dados referentes a consultas,

[2]https://www.linx.com.br/clinicweb

[3]https://www.gestaods.com.br

[4]https://lps.iclinic.com.br/sistema-para-clinicas-e-consultorios

pacientes e profissionais para uma clínica de pequeno porte.

### 2.1. Objetivos específicos

Para alcançar o objetivo geral, o trabalho teve como objetivos específicos:

1. Realizar o levantamento de requisitos do sistema;
2. Criar um banco de dados para controle dos dados da clínica; e
3. Desenvolver o sistema utilizando tecnologias para web.

O capítulo a seguir conceitua as tecnologias e ferramentas utilizadas para se alcançar os objetivos do trabalho.

## 3. Fundamentação Teórica

A tecnologia tem sido uma das grandes aliadas do setor da saúde, uma vez que esta permite uma gestão mais eficiente e otimizada dos processos clínicos. É nesse contexto que as soluções de agendamento para clínicas e consultórios médicos surgem como uma alternativa para melhorar a organização e qualidade do atendimento ao paciente. O produto tecnológico, fruto do presente trabalho, utilizou-se das tecnologias PHP, JavaScript, HTML, CSS, *frameworks* Code Igniter e Bootstrap, e o Sistema de Gerenciamento de Bancos de Dados (SGBD) MySQL, as quais são detalhados nesta seção.

A escolha das tecnologias e ferramentas utilizadas no desenvolvimento do produto teve como base o fato de serem amplamente difundidas, e com uma grande comunidade de desenvolvedores presente na rede mundial de computadores, o que garante uma maior facilidade de desenvolvimento e manutenção do sistema.

### 3.1. Linguagens para desenvolvimento WEB

Segundo Gabriel [2022], *front-end* é a camada da aplicação web que se comunica com o usuário, usando HTML, CSS e JavaScript para criar uma interface gráfica. O *front-end* envia e recebe dados do *back-end*, que é a camada que processa as informações. O navegador é o programa que interpreta o código do *front-end* e o exibe na tela, seguindo os padrões do *World Wide Web Consortium* (W3C). As tecnologias *front-end* utilizadas no presente trabalho são detalhadas a seguir.

O HTML, do inglês *Hypertext Markup Language* (Linguagem de Marcação Hipertexto) é uma linguagem utilizada para produzir páginas na web. Ela é composta por uma série de marcações que são interpretadas pelos navegadores para exibir o conteúdo da página. As aplicações da linguagem HTML são diversas e incluem a criação de sites, blogs, portais de notícias, lojas virtuais e outras aplicações web.[da Silva Neto, sd]

A linguagem CSS, do inglês *Cascade Style Sheet* (Folha de Estilos em Cascata), é muito utilizada junto com HTML ou XHTML, e representa diversas possibilidades para a formatação de uma página web. A linguagem CSS ajuda a editar, alinhar, remover e trabalhar no espaço entre elementos de uma página. Além disso, a linguagem CSS permite criar layouts responsivos, que se adaptam a diferentes tamanhos de tela e dispositivos, e personalizar o design de um site com cores, fontes, imagens e animações. Com a linguagem CSS também facilita-se a manutenção e a padronização do código, pois separa-se o conteúdo da apresentação visual. [Copes, 2023]

PHP “é uma linguagem de *script open source* de uso geral, muito utilizada, e especialmente adequada para o desenvolvimento web e que pode ser embutida dentro do HTML” [PHP, 2023]. Segundo Welling [2005], o PHP tem conexões nativas disponíveis para muitos sistemas de banco de dados. Além do MySQL, é possível conectar-se diretamente à bancos de dados PostgreSQL, Oracle, dbm, FilePro, HyperWave, Informix, InterBase, Sybase, entre outros.

Com a linguagem PHP é possível alternar entre as linguagens, embutindo o código de uma linguagem diferente dentro de um bloco PHP. Por exemplo, é possível embutir código HTML, JavaScript ou até mesmo código de outras linguagens de programação, como Ruby ou Perl, dentro de um bloco PHP como mostra o pequeno trecho de código para montagem da tabela dinâmica da Figura 1, onde todo o código que está entre “<?php ” e “ ?>” é código PHP inserido junto ao código HTML.

```
<tbody>
  <?php foreach ($pesquisarconsultas as $consulta) : ?>
    <tr>
      <td id="nome-<?= $consulta['Id_Agendamento']; ?>"><?= $consulta['Nome_Paciente'];
      ?></td><td><?php echo $consulta['Nome_Profissional']; ?></td>
      <td><?php echo $consulta['Tipo_Consulta']; ?></td>
      <td><?php echo date_format(date_create($consulta['agendamento']), 'd/m/Y'); ?></td>
      <td><?php echo date_format(date_create($consulta['horario']), 'H:i'); ?></td>
      <td><?php echo 'R$ ' . number_format($consulta['Valor'], 2, ',', '.'); ?></td>
      <td <?php if ($consulta['Estado'] == 'realizada') echo "style='color:green'";
          else echo "style='color:red'" ?>><?php echo $consulta['Estado']; ?></td>
      <td style="white-space: nowrap;">
        <a class="btn btn-primary" href="/sinteag/public/EditarConsulta/
        <?php echo $consulta['Id_Agendamento'];
        ?>"><i class="bi bi-pencil-square"></i></a>
        <button id="btnExcluir" onclick="confirmaExclusao(
          <?= $consulta['Id_Agendamento'];
          ?>)" class="btn btn-danger"><i class="bi bi-trash"></i></button>
      </td>
    </tr>
  <?php endforeach; ?>
</tbody>
```

**Figura 1. Exemplo de código PHP com HTML para montagem de tabela dinâmica**

A linguagem PHP possui integração simplificada com o banco de dados MySQL (detalhado a seguir). Além disso, cabe ressaltar que a documentação da linguagem PHP[5] é simples, objetiva, e bem extensa, podendo ser acessada online.

Segundo Flanagan [2011], JavaScript é uma linguagem de programação de alto nível, dinâmica e interpretada, amplamente utilizada para desenvolvimento web. Ela permite a criação de páginas interativas e dinâmicas por meio da manipulação do Modelo de Objetos do Documento (DOM), interação com o usuário e criação de animações. Além disso, JavaScript é uma linguagem multiplataforma, ou seja, pode ser executada em diversos sistemas operacionais e navegadores.

[5]http://php.net

Uma das vantagens da linguagem JavaScript é a compatibilidade com todos os navegadores modernos, que assegura o funcionamento adequado do código em diferentes ambientes. Outra vantagem é a versatilidade da linguagem, que possibilita criar aplicações complexas e interativas, oferecendo uma ótima experiência ao usuário. A reutilização de código e a grande quantidade de bibliotecas e *frameworks* disponíveis também são benefícios que agilizam o desenvolvimento e facilitam a manutenção do sistema [Resig, 2007].

O CodeIgniter[6] é um *framework* de desenvolvimento de aplicações PHP que é considerado uma caixa de ferramentas, proporcionando o desenvolvimento mais rápido de aplicações com um excelente conjunto de bibliotecas para tarefas comuns, além de interfaces simples e uma estrutura lógica para o acesso dessas bibliotecas. A utilização deste *framework* permite uma redução significativa no número de linhas de código necessárias para uma tarefa, comparado com outras soluções de desenvolvimento [DevMedia, 2023a].

O CodeIgniter possui classes predefinidas que podem ser combinadas e estendidas para construirmos nossas aplicações, permitindo economizar tempo de codificação. Além disso, ele possui uma ampla biblioteca de funções, chamadas de *helpers*, agrupadas de acordo com suas finalidades. O CodeIgniter é gratuito, leve e rápido. Utiliza a arquitetura *Model*, *View*, *Controler* (MVC), gera URL (Localizador Uniforme de Recursos) limpas e amigáveis a sites de busca, e é altamente extensível através do uso de suas próprias bibliotecas, *helpers* e extensões de classes [DevMedia, 2023a].

A arquitetura MVC é um padrão que separa a aplicação em três camadas: *Model*, *View* e *Controller*. O *Model* (modelo) é responsável pela manipulação dos dados, a *View* (visualização) é responsável pela interação do usuário, e o *Controller* (controlador) é responsável pela comunicação entre as duas camadas. O MVC ajuda na redução de acoplamento e promove o aumento de coesão nas classes do projeto, e também facilita a alteração do layout sem afetar os dados, e vice-versa [DevMedia, 2023b].

O padrão MVC funciona da seguinte forma: quando o usuário faz uma requisição, o *Controller* recebe essa requisição e delega para o *Model* a tarefa de acessar e processar os dados necessários. O *Model* retorna os dados para o *Controller*, que por sua vez envia os dados para a *View*. A *View* então gera a interface do usuário, que é apresentada ao usuário. O *Controller* é o intermediário entre o *Model* e a *View*, fazendo a ligação entre eles.

O Bootstrap é um *framework front-end* e de código-aberto criado por Mark Otto e Jacob Thornton para tornar o desenvolvimento web mais rápido e prático. Este *framework* oferece uma variedade de *templates* baseados em HTML e CSS para diferentes funções e componentes, como navegação, grades, carrosséis de imagens e botões. Este *framework front-end* permite criar sites responsivos, otimizados para qualquer tamanho de tela, desde dispositivos móveis até computadores potentes. Isso permite que os desenvolvedores criem várias versões do mesmo site em um mesmo endereço/documento para diferentes tipos de tela e não limite a audiência do site. Devido à sua popularidade, surgiram várias comunidades do Bootstrap, onde os desenvolvedores e designers podem trocar experiências, conhecimentos e discutir as últimas atualizações do *framework* [hostinger,

[6]https://codeigniter.com/

2023].

Optou-se pelo uso do Sistema de Gerenciamento de Banco de Dados (SGBD) MySQL porque é um sistema aberto e gratuito, que faz parte do grupo de aplicações LAMP (Linux + Apache + MySQL + PHP) de desenvolvimento de aplicações web sem custos. Além disso, o MySQL é um dos gerenciadores de banco de dados gratuitos mais populares e confiáveis utilizados atualmente na internet [MySQL, 2023].

Segundo PortoFacil [2023], o MySQL tem as seguintes vantagens:

1. Software muito popular e confiável, assim como sua flexibilidade e escalabilidade;
2. Possui alta performance e disponibilidade, além de suporte transacional extremamente robusto; e
3. Oferece código estável com várias funcionalidades.

Tendo definidas as tecnologias utilizadas, o próximo capítulo traz o passo-a-passo percorrido para realizar o presente trabalho.

## 4. Metodologia

A metodologia de pesquisa utilizada neste trabalho é a de pesquisa descritiva, que conforme Paslavik et al. [2021], tem por objetivo observar, registrar, analisar e interpretar os fatos do mundo físico e humano, sem interferir ou manipular os mesmos. Assim foi realizada entrevista com profissional da área para se entender as necessidades e estudo de casos ligados a clínica e sua dinâmica de atendimento.

O *software* foi desenvolvido seguindo uma metodologia iterativa e incremental, que é uma forma comum de engenharia de software. Essa metodologia consiste em dividir o desenvolvimento em ciclos curtos e repetitivos, chamados de iterações. Em cada iteração, são realizadas atividades de planejamento, análise, design, codificação, teste e avaliação [Mauricio, 2012]. O propósito é desenvolver o produto de forma gradual, com *feedback* constante e melhorias incrementais. Os requisitos foram definidos e modelados no início do projeto, e depois implementados e testados em ciclos sucessivos de desenvolvimento.

O primeiro passo foi o levantamento de requisitos, o qual foi realizado por meio de entrevistas com um *stakeholder* do projeto. A partir dessa entrevista, foi possível identificar os requisitos funcionais do sistema, tais como cadastro, pesquisa, listagem e edição de pacientes, profissionais e usuários do sistema, bem como as funcionalidades de agendamento de consultas.

Seguindo o levantamento de requisitos realizado e validado no inicio do projeto, criou-se o diagrama de entidades e relacionamentos, e em seguida o modelo físico do banco de dados, no SGBD MySQL.

Em seguida, foram criadas as telas do sistema, utilizando a linguagem HTML, CSS e os *framework* Bootstrap e CodeIgniter. Foram desenvolvidas telas de cadastro, pesquisa e edição de pacientes, profissionais e usuários, bem como telas de agendamento. As telas foram projetadas com base nos requisitos levantados, de modo a tornar o sistema fácil de usar e intuitivo para os usuários.

Com as telas prontas e validadas, foram implementados os códigos referentes aos requisitos levantados, utilizando a linguagem PHP, JavaScript e o *framework* CodeIgniter.

Foram desenvolvidas funções de cadastro, pesquisa, edição e listagens de pacientes, profissionais e usuários, bem como funções de agendamento. Esses códigos foram integrados com o banco de dados, de modo a permitir o armazenamento e recuperação de dados.

Por fim, foram realizados testes do sistema, buscando identificar e corrigir eventuais problemas e inconsistências. Após a conclusão dos testes, o sistema foi implantado em um ambiente de produção, para uso efetivo pelos usuários.

Dessa forma, a metodologia utilizada neste trabalho de conclusão de curso permitiu a implementação de um sistema de agendamentos para consultas médicas, que atendeu às necessidades dos usuários e dos *stakeholders* envolvidos no projeto. O capítulo a seguir relata todo o processo de desenvolvimento do produto tecnológico.

## 5. Desenvolvimento

Visando a elaboração do projeto de um sistema para agendamentos de consultas web, realizou-se o levantamento de requisitos por meio de entrevistas com os *stakeholders*, e durante as entrevistas foram questionados os procedimentos e a dinâmica de atendimento da clínica, o fluxo, a quantidade de pacientes e profissionais para atendimento, o horário de atendimento, os usuários e seus níveis de instrução, as regras de negócio impostas pelo proprietário da clínica, bem como a existência de legislações que pudessem impor restrições ou a necessidade de implementação de alguma funcionalidade específica.

Após a reunião, considerações e sugestões, levantou-se os seguintes requisitos necessários para atender às necessidades do Sistema de agendamento de consultas, armazenamento e gerenciamento de dados da clínica:

1. Autenticação de login para acesso e início do sistema;
2. Cadastro dos profissionais que realizarão os atendimentos, dos pacientes que serão atendidos e do agendamento das consultas a serem realizadas com as seguintes informações;
3. O cadastro de pacientes deverá conter as seguintes informações: nome do paciente, número de CPF, data de nascimento, endereço completo (com rua, número, complemento, bairro, CEP, cidade, UF), número de telefone e campo para observações e diagnósticos;
4. O cadastro de profissionais deverá conter as seguintes informações: nome do profissional, especialidade, número de CPF, data de nascimento, sexo, endereço completo (com rua, número, complemento, bairro, CEP, cidade, UF) e número de telefone;
5. O cadastro de agendamentos deverá conter as seguintes informações: nome do paciente, nome do profissional, tipo de consulta (consultório ou domiciliar), valor da consulta, data, hora e status (aguardando ou realizada);
6. Função para edição dos cadastros que possuam alguma inconsistência ou que seja necessário efetuar alterações;
7. Emissão de listagens com relação aos pacientes, profissionais e consultas;
8. O sistema não deve permitir o cadastro de agendamento para o profissional se o mesmo já possuir agendamento para o mesmo dia e hora;
9. O sistema não deve permitir a exclusão do cadastro de paciente ou profissional que possua registro de agendamento;
10. A aplicação deve ser web para que o usuário possa acessá-la de qualquer lugar; e

11. A aplicação deve se adaptar aos mais variados tamanhos de tela, possibilitando sua exibição nas diversas plataformas existentes (responsividade).

Com a validação dos requisitos levantados, das telas e a elaboração do diagrama relacional para criação do banco de dados, deu-se início à construção da aplicação utilizando as tecnologias mencionadas na Seção 3, e conforme detalhado a seguir.

Para facilitar a utilização do sistema, as telas de pesquisa devem retornar apenas os dados necessários para a atividade de registro de agendamento e consulta de dados fundamentais do paciente e do profissional, conforme descritos a seguir:

1. Pesquisa Pacientes: Nome do paciente, CPF e Telefone;
2. Pesquisa Profissional: Nome do profissional, CPF, telefone e sexo;
3. Pesquisa Agendamentos: Nome do profissional, Nome do Paciente, Local da Consulta, Data, hora, valor e status;

Com base nos requisitos levantados, realizou-se a modelagem do banco de dados, resultando em um Diagrama de Entidade-Relacionamento (DER) mostrado na Figura 2, com quatro tabelas: Pacientes, Profissionais, Agendamentos e Usuários. Foram definidas as tabelas, colunas e relações entre elas, de modo que o banco de dados pudesse armazenar corretamente as informações do sistema. Os relacionamentos foram estabelecidos da seguinte maneira:

1. O Relacionamento pacientes para agendamentos com a cardinalidade de um para muitos, uma vez que o mesmo paciente poderá ser atendido inúmeras vezes;
2. O Relacionamento Profissional para agendamentos com a cardinalidade um para muitos, uma vez que o mesmo profissional poderá realizar quantos atendimentos se fizerem necessários.

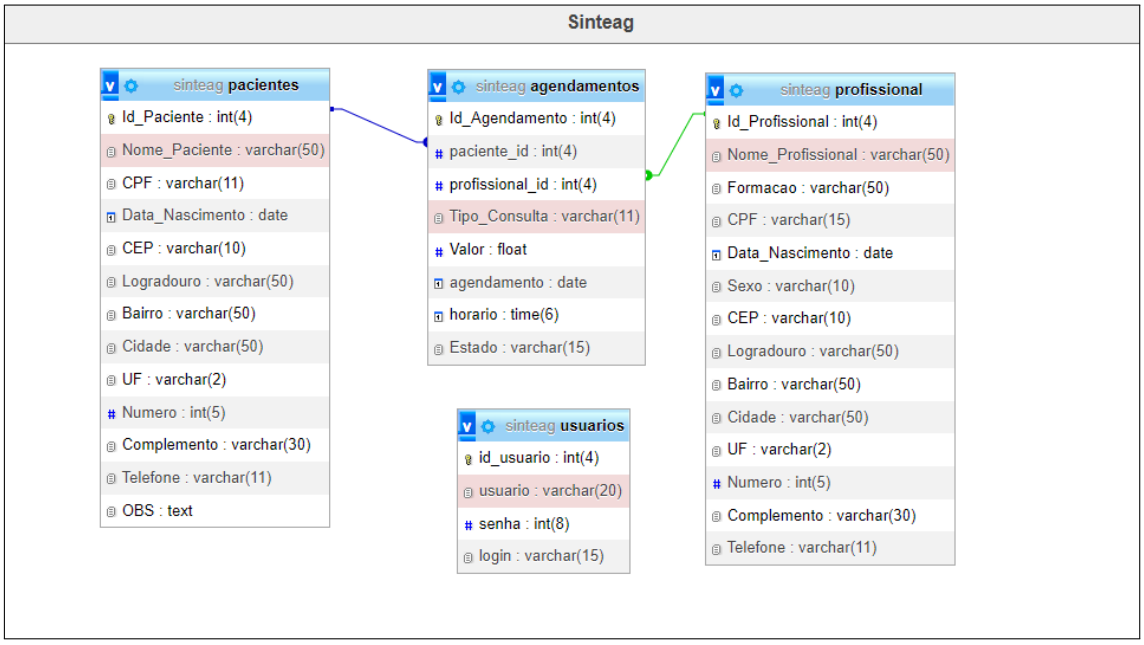


**Figura 2. Modelagem de banco de dados relacional**

Seguindo o fluxo de desenvolvimento do projeto, depois de aprovadas as telas de acordo com o levantamento de requisitos, deu-se continuidade ao desenvolvimento, e implementou-se as funcionalidades de validações inerentes a cada uma de acordo com a iteratividade que o usuário terá com o sistema.

Todas as páginas do sistema exibem o menu suspenso de cadastro das classes Paciente, Profissional, Consultas e Usuários, e o menu suspenso de Pesquisa das classes Paciente, Profissional e Consultas. Elas também possuem uma *label* que é uma legenda para um item em uma interface de usuário, indicadora de qual usuário está logado e um botão de Sair (*Logoff*). Além disso, há outra *label* com o nome do sistema (SINTEAG) que, ao ser clicada, retorna à página principal.

Na construção das telas que possuem tabelas, foi utilizado o *plugin DataTables*, que junto à biblioteca de JavaScript chamada *jQuery*, é uma ferramenta altamente flexível, construída sobre os fundamentos do aprimoramento progressivo, que adiciona todos esses recursos avançados a qualquer tabela HTML. Esta biblioteca é usada para controle e manipulação interativa de tabelas dinâmicas. O *DataTables* carrega tabelas facilmente, e o site fica com um layout amigável, além de funções de busca, ordenação e paginação [DataTables, 2023].

Atendendo a um requisito, a tela inicial o sistema busca informações do banco de dados e apresenta uma tabela com todas as consultas agendadas para o dia, com os respectivos dados dos clientes, para confirmação do comparecimento dos mesmos, como pode ser visualizado na Figura 3.


**Figura 3. Tela inicial do sistema, com as consultas agendadas para o dia**

Todas as telas de pesquisa que mostram uma listagem possuem as funcionalidades de pesquisa para facilitar a busca do usuário, quando a relação já possuir uma grande quantidade de dados. Essas funcionalidades podem ser realizadas através do campo de entrada de pesquisa, pela paginação, ou na ordenação por qualquer dado que retorne do banco de dados, como profissional, paciente, CPF, etc.

Conforme a lista de requisitos, foram construídas quatro telas de cadastro, sendo elas: Profissional, Paciente, Consulta e Usuário. Foram construídas também três telas de pesquisa, sendo elas: Pesquisa de Profissional, Pesquisa de Paciente e Pesquisa de Consulta. Além disso, foram criadas quatro telas de edição, uma para cada tipo de cadastro. Foram implementadas as seguintes funcionalidades em todas as telas de cadastro:

1. Conexão com o banco de dados;
2. Captura dos dados inseridos nos respectivos formulários;
3. Inserção dos dados na respectiva tabela do banco de dados através do comando *insert* da linguagem SQL; e
4. Exibição da confirmação de inserção realizada através do uso da biblioteca JavaScript Sweetalert2 (vide Figura 4).

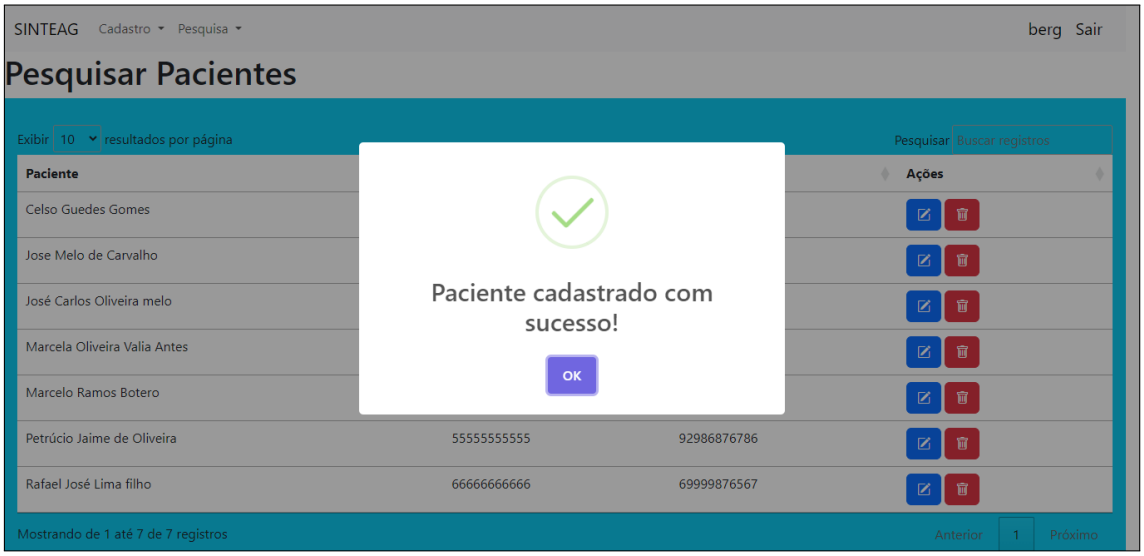


**Figura 4. Mensagem de sucesso ao cadastrar um paciente**

A Figura 4 mostra uma mensagem de confirmação do cadastro do paciente, quando a mesma aconteceu satisfatoriamente, ou seja, todas as validações passaram e os dados foram inseridos no banco de dados.

Nas telas de pesquisa para a manipulação dos registros (exemplo na Figura 5) não se faz necessário mostrar todos os dados registrados. O sistema busca apenas as informações solicitadas no levantamento de requisitos. Para cada linha da tela de pesquisa foram criados os botões de editar (em cor azul), e excluir (em cor vermelha), para que assim esta funcionalidade fosse direcionada ao "id" correspondente ao elemento da tabela, e o uso das cores para que o uso da funcionalidade seja intuitivo.

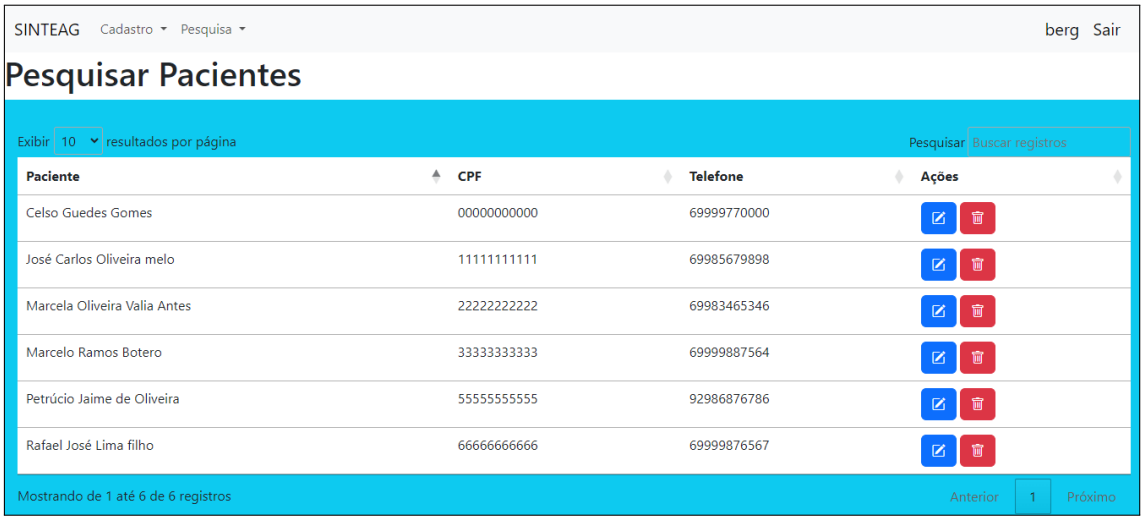


**Figura 5. Exemplo de tela de pesquisa de pacientes**

A funcionalidade do botão editar foi projetada para mostrar as telas de edição, seja de Profissional, Paciente ou Consulta, já preenchendo o respectivo formulário ou tabela com as informações existentes no banco de dados referente ao elemento que se deseja alterar. Tais dados são solicitados ao SGBD através de um comando SQL e, após a alteração de dados, é informado se houve algum erro, retornando à tela de edição, ou se a edição foi realizada com sucesso, retornando à tela de pesquisa, que é preenchida já com as atualizações, conforme o trecho de código disposto na Figura 6.

```php
public function index()
{
    $db = \config\Database::connect();
    $query = $db->query("
    SELECT a.Id_Agendamento, a.Tipo_Consulta , p.Nome_Paciente,
    pr.Nome_Profissional, a.agendamento, a.Valor, a.horario, a.Estado
    FROM agendamentos as a
    join pacientes p ON a.paciente_id = p.Id_Paciente
    join profissional pr ON a.Profissional_Id = pr.Id_Profissional");
    $resultado = $query->getResultArray();

    $data = [
        'titulo' => 'Pesquisar Consultas',
        'pesquisarconsultas' => $resultado,
        'sucesso' => $this->session->getFlashdata('sucesso'),
        'erro' => $this->session->getFlashdata('erro')
    ];

    return view('pesquisarconsultas', $data);
}
```

**Figura 6. Exemplo de consulta SQL e retorno de informações**

A funcionalidade de exclusão é aplicada com cautela, visando evitar que o usuário acidentalmente delete um registro de forma equivocada. Para evitar essa falha, um aviso de confirmação de exclusão é exibido antes de executar essa ação vide Figura 7.


**Figura 7. Confirmação de exclusão**

Cada tabela do banco de dados possui identificadores que permitem a relação entre si. Esses identificadores, também conhecidos como chaves primárias em suas próprias tabelas e chaves estrangeiras se relacionados em outras tabelas, neste sistema são utilizados para incluir informações de profissionais e pacientes em uma tabela de consulta. É importante ressaltar que a integridade do banco deve ser mantida, ou seja, não se deve excluir um elemento que esteja relacionado em outra tabela. Isso ocorre porque ao consultar um registro de consulta o sistema não terá as informações tanto do paciente quanto do profissional relacionado, portanto gerando uma quebra de integridade.

Um exemplo prático de quebra de integridade seria a exclusão de um paciente ou profissional que está relacionado a uma consulta. Caso um deles seja excluído, o sistema não será capaz de fornecer informações completas sobre a consulta, deixando o banco de dados em um estado incorreto e potencialmente inconsistente. Desta forma, é necessário garantir a integridade referencial e implementar mecanismos, como restrições de integridade, para prevenir a exclusão acidental ou não autorizada de elementos relacionados.

Para garantir a integridade dos dados, foi implantada a validação de verificação se há chave estrangeira na tabela que possa comprometer a integridade do banco de dados. Se esta verificação for verdadeira, não será permitida a exclusão e será emitido o aviso e o motivo da negativa de exclusão como mostra a Figura 8, retornando à tela de pesquisa. Caso as validações permitam, o registro será excluído e será emitido o aviso de exclusão com sucesso.


**Figura 8. Aviso de erro de chave estrangeira**

Foram realizados testes de funcionamento de integração com o banco de dados e identificou-se que o registro do CPF, que no SGBD é configurado para 11 (onze) caracteres, caso o usuário inserisse uma quantidade maior, o SGBD registrava apenas a quantidade de caracteres definida (os onze primeiros dígitos). Para solucionar tal problema, foi implementada a validação dentro da aplicação, que só aceita e envia o número exato de caracteres, emitindo o aviso no próprio campo de entrada, e só permitindo o envio do cadastro após a correção pelo usuário. Da mesma forma, foram executados testes com todos os campos e implementadas validações onde se apresentaram inconsistências no sistema, e em campos obrigatórios, como por exemplo o campo nome com menos de três caracteres.

Assim como as verificações de integridade do banco de dados, é igualmente importante contar com rotas bem definidas e controles de acesso para um sistema como o proposto. Uma rota é o caminho que o sistema deve buscar, baseado no endereço de navegação acessado pelo usuário, para exibir a página web correspondente de maneira correta. Após finalizar a criação e implementação do projeto, foi elaborado o código disposto na Figura 9, onde a se inseriu uma condicional de validação para inicio de sessão (destacado em cor laranja) e, somente após satisfeita tal condicional (*if* destacado em cor verde), será permitido o acesso a todas as rotas e funcionalidades do sistema. Tal funcionalidade tem por objetivo a proteção dos dados, uma vez que só poderão ser acessados por usuário efetivamente autorizado e autenticado no sistema. Não satisfeita a condição de autenticação (*else* em destacado em cor vermelha), o sistema retorna a tela de login.

Todos os envios de dados foram realizados pelo método POST, que é uma forma de enviar dados para um servidor web, que podem ser processados por um *script* PHP. O método POST envia os dados no corpo da requisição, o que torna a URL mais limpa e permite enviar dados mais complexos, como arquivos ou *arrays*. O método POST também é mais adequado para enviar dados sensíveis, como senhas ou informações pessoais, pois eles não ficam expostos na URL.

Neste capítulo foi apresentado o desenvolvimento de um sistema de agendamento de consultas, que é o objetivo principal do presente trabalho. Foram detalhadas as etapas de levantamento de requisitos, modelagem e implementação de banco de dados MySQL, construção do sistema, controle de acesso via rotas e funcionalidades de agendamentos, listagem de pacientes, profissionais e consultas.

```
$session = \Config\Services::session();
$routes->get('/Login', 'Login::index');
$routes->post('/Login', 'Login::autenticar');

if ($session->get('logado')) {
    $routes->get('/logout', 'Login::deslogar');
    $routes->get('/', 'Home::index');
    $routes->get('/CadastrarPacientes','CadastrarPacientes::index');
    $routes->post('/CadastrarPacientes/cadastrar','CadastrarPacientes::cadastrar');
    $routes->get('/CadastrarProfissionais','CadastrarProfissionais::index');
    $routes->post('/CadastrarProfissionais/cadastrar','CadastrarProfissionais::cadastrar');
    $routes->get('/CadastrarUsuarios', 'CadastrarUsuarios');
    $routes->post('/CadastrarUsuarios/cadastrar','CadastrarUsuarios::cadastrar');
    $routes->get('/CadastrarConsultas', 'CadastrarConsultas::index');
    $routes->post('/CadastrarConsultas/cadastrar', 'CadastrarConsultas::cadastrar');
    $routes->get('/EditarConsulta/excluir/(:any)', 'EditarConsulta::excluir/$1');
    $routes->post('/EditarConsulta/atualizar/(:any)', 'EditarConsulta::atualizar/$1');
    $routes->get('/EditarConsulta/(:any)', 'EditarConsulta::index/$1');
    $routes->post('/EditarPaciente/atualizar/(:any)', 'EditarPaciente::atualizar/$1');
    $routes->get('/PesquisarPacientes', 'PesquisarPacientes::index');
    $routes->get('/PesquisarProfissionais', 'PesquisarProfissionais');
    $routes->get('/PesquisarConsultas', 'PesquisarConsultas');
    $routes->get('/MovimentoCaixa', 'MovimentoCaixa');
    $routes->get('/RelatorioDeCaixa', 'RelatorioDeCaixa');
    $routes->get('/EditarPaciente/index/(:any)', 'EditarPaciente::index/$1');
    $routes->get('/EditarPaciente/excluir/(:any)', 'EditarPaciente::excluir/$1');
    $routes->get('/EditarProfissional/index/(:any)', 'EditarProfissional::index/$1');
    $routes->post('/EditarProfissional/atualizar/(:any)', 'EditarProfissional::atualizar/$1');
    $routes->get('/EditarProfissional/excluir/(:any)', 'EditarProfissional::excluir/$1');
} else {
    $routes->get('/', 'Login::index');
}
```

**Figura 9. Controle de acesso com rotas**

No próximo capítulo, apresenta-se a conclusão do trabalho, destacando as principais contribuições, limitações e sugestões para trabalhos futuros.

## 6. Conclusão

Com base nos dados coletados e na metodologia adotada, foi possível desenvolver um sistema web para agendamentos de consultas em clínicas de pequeno porte onde foi realizado o levantamento de requisitos do sistema, criado um banco de dados e desenvolvido o produto tecnológico chamado de SINTEAG, utilizando tecnologias de desenvolvimento para web.

A linguagem de programação PHP, juntamente com o *framework* CodeIgniter e o SGBD MySQL, se mostraram tecnologias de fácil uso, flexíveis e robustas. Além disso, o JavaScript proporcionou o aprimoramento da experiência do usuário, tornando a interface do sistema mais interativa e amigável.

O sistema implementado permite o cadastro e armazenamento de dados pessoais dos pacientes, informações dos profissionais atendentes, bem como o registro de todas as consultas realizadas e agendadas, conforme os requisitos apontados. Disponibiliza listagens completas e atualizadas dos pacientes, profissionais e consultas, facilitando a

gestão e o acompanhamento das atividades da clínica.

Nesta pesquisa, evidenciou-se que as clínicas de pequeno porte precisam de mais do que apenas um cadastro de agendamentos para aprimorar a gestão das suas atividades. Um dos pontos que merece atenção e aprimoramento no SINTEAG é o controle de movimentação financeira, englobando registro de entradas e saídas, contas a receber e contas a pagar. Portanto, para trabalhos futuros, propõe-se a pesquisa e desenvolvimento de um módulo dedicado exclusivamente a esse controle bem como a integração do SINTEAG com ferramentas de agenda para envio do arquivo ao cliente.

Com isso, conclui-se que o sistema de agendamentos desenvolvido atende aos requisitos levantados, e portanto às demandas de clínicas de pequeno porte, contribuindo para a organização e o controle das consultas, melhorando a eficiência do atendimento aos pacientes e proporcionando uma experiência satisfatória para todos os envolvidos, cumprindo assim os objetivos deste trabalho.

## Referências


BRASIL (2023). Síntese de casos, óbitos, incidência e mortalidade. https://covid.saude.gov.br/ [Acesso em 08/05/2023].

CONASS (2023). Painel Conass Covid-19 completa mil dias de divulgação diária! https://www.conass.org.br/painelconasscovid19/ [Acesso em 08/05/2023].

Copes, F. (2023). Manual do CSS- Um guia pratico de CSS para desenvolvedores. https://www.freecodecamp.org/portuguese/news/manual-do-css-um-guia-pratico-de-css-para-desenvolvedores/ [Acesso em 03/11/2023].

da Silva Neto, P. C. (s.d). Desenvolvimento de Aplicações Web. https://educapes.capes.gov.br/bitstream/capes/642806/2/DAW02.pdf [Acesso em 18/06/2023].

DataTables (2023). DataTables. https://datatables.net/ [Acesso em 09/11/2023].

DevMedia (2023a). Introdução ao framework PHP CodeIgniter. https://www.devmedia.com.br/introducao-ao-framework-php-codeigniter/27346 [Acesso em 18/06/2023].

DevMedia (2023b). O que é MVC? https://www.devmedia.com.br/mvc/ [Acesso em 03/11/2023].

Flanagan, D. . (2011). *JavaScript: TheDefinitive Guide 6th Edition*. O'Reilly Media, Inc.

Gabriel, P. A. (2022). *Front-End: Curso Completo de HTML, CSS e JavaScript*. Tech Stuff House.

hostinger (2023). O que é o Bootstrap? https://www.hostinger.com.br/tutoriais/o-que-e-bootstrap [Acesso em 03/11/2023].

Mauricio, Aniche, M. D. D. (2012). *Métodos Ágeis de Desenvolvimento de Software*. Editora Bookman.

MoneyLab (2021). Pós-pandemia: Ti assume papel estratégico nas empresas e na competitividade. https://www.infomoney.com.br/patrocinados/empresas-e-tecnologia/pos-pandemia-ti-assume-papel-estrategico-nas-empresas-e-na-competitividade/ [Acesso em 18/06/2023].

Morais, C. R. (2020). Saiba quais são as vantagens do agendamento online para clínicas e consultórios médicos. https://www.medplus.com.br/saiba-quais-sao-as-vantagens-do-atendimento-online-para-clinicas/ [Acesso em 18/06/2023].

MySQL (2023). The world's most popular open-source database. https://www.mysql.com/ [Acesso em 19/06/2023].

Paslavik, D., Santos, R., M., S., and Oliveira, L. (2021). *Metodologia da pesquisa científica: guia prático para a elaboração de trabalhos acadêmicos*. Editora da Universidade de São Paulo.

PHP (2023). O que é o PHP? https://www.php.net/manual/pt_BR/intro-whatis.php [Acesso em 19/06/2023].

Pinochet, L. H. C. (2011). Tendências de tecnologia de informação na gestão da saúde. *O mundo da Saúde*, 35(4):382–394.

PortoFacil (2023). The world's most popular open-source database. www.portofacil.net/banco-de-dados-mariadb-e-mysql.html [Acesso em 19/06/2023].

Resig, J. (2007). *Técnicas profissionais de JavaScript*. Apress.

Welling, Luke e Thomson, L. (2005). Php e mysql desenvolvimento web, 3º edição. *Campus, Rio de Janeiro*.