

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA  
CAMPUS CAJAZEIRAS  
CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE  
SISTEMAS**

**HEALTH CLINIC: APLICAÇÃO WEB NA SAÚDE, AMPLIANDO O  
ACESSO E A COMUNICAÇÃO.**

**RAFAEL BARROSO DE SANTANA**

**Cajazeiras - PB  
2024**

**RAFAEL BARROSO DE SANTANA**

**HEALTH CLINIC: APLICAÇÃO WEB NA SAÚDE, AMPLIANDO O ACESSO E A  
COMUNICAÇÃO.**

Trabalho de Conclusão de Curso apresentado junto ao Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia da Paraíba - Campus Cajazeiras, como requisito à obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Orientador

Prof. Me. João Paulo Freitas de Oliveira.

Prof. Dr. Leandro Luttiane da Silva Linhares. -  
Coorientador

**Cajazeiras - PB  
2024**

IFPB / Campus Cajazeiras  
Coordenação de Biblioteca  
Biblioteca Prof. Ribamar da Silva  
Catalogação na fonte: Cícero Luciano Félix CRB-15/750

S232h Santana, Rafael Barroso de.  
*Health clinic* : aplicação *web* na saúde, ampliando o acesso e a comunicação / Rafael Barroso de Santana.– 2024.

45f. : il.

Trabalho de Conclusão de Curso (Tecnólogo em Análise e Desenvolvimento de Sistemas) - Instituto Federal de Educação, Ciência e Tecnologia da Paraíba, Cajazeiras, 2024.

Orientador(a): Prof. Me. João Paulo Freitas de Oliveira.  
Coorientador(a): Prof. Dr. Leandro Luttiane da Silva Linhares.

1. Desenvolvimento de sistemas. 2. Plataforma web Health clinic.  
3. Sistema de gestão. 4. Gestão de serviços em saúde. I. Instituto Federal de Educação, Ciência e Tecnologia da Paraíba. II. Título.

IFPB/CZ

CDU: 004.4(043.2)



MINISTÉRIO DA EDUCAÇÃO  
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA  
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA

RAFAEL BARROSO DE SANTANA

**HEALTH CLINIC: APLICAÇÃO WEB NA SAÚDE, AMPLIANDO O  
ACESSO E A COMUNICAÇÃO**

Trabalho de Conclusão de Curso apresentado junto ao Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia da Paraíba - Campus Cajazeiras, como requisito à obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Orientador

Prof. Me. João Paulo Freitas de Oliveira

Aprovada em: **28 de Fevereiro de 2024.**

Prof. Me. João Paulo Freitas de Oliveira - Orientador

Profa. Esp. Asheley Emmy Lacerda Alves - Avaliadora

IFPB - Campus Cajazeiras

Prof. Tecg.o. Antônio Ricart de Oliveira Medeiros - Avaliador

IFPB - Campus Cajazeiras

Documento assinado eletronicamente por:

- **Joao Paulo Freitas de Oliveira, PROF ENS BAS TEC TECNOLOGICO-SUBSTITUTO**, em 28/02/2024 16:46:09.
- **Antonio Ricart Jacinto de Oliveira Medeiros, PROF ENS BAS TEC TECNOLOGICO-SUBSTITUTO**, em 28/02/2024 16:49:06.
- **Asheley Emmy Lacerda Alves, PROF ENS BAS TEC TECNOLOGICO-SUBSTITUTO**, em 01/03/2024 09:46:04.

Este documento foi emitido pelo SUAP em 28/02/2024. Para comprovar sua autenticidade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifpb.edu.br/autenticar-documento/> e forneça os dados abaixo:

Código 538714  
Verificador: 01b781ca86  
Código de Autenticação:



Rua José Antônio da Silva, 300, Jardim Oásis, CAJAZEIRAS / PB, CEP 58.900-000  
<http://ifpb.edu.br> - (83) 3532-4100

## **AGRADECIMENTOS**

A expressão de minha gratidão se direciona, em especial, aos meus dedicados pais, Francisca Barroso e Marciano Ferreira, que não apenas me apoiaram, mas também me incentivaram ao longo desta jornada acadêmica. Sou profundamente grato aos meus orientadores, João Paulo e Leandro Luttiane, pelos valiosos ensinamentos e conhecimentos compartilhados, fundamentais para a conclusão deste trabalho. Manifesto meu reconhecimento aos queridos amigos David Beserra, Lucas Tavares, Kauanny Vieira e Reyson Renan, cuja companhia tornou esta longa caminhada repleta de momentos especiais e aprendizados enriquecedores. Não poderia deixar de agradecer à minha futura esposa, Giovanna de Lucena, por todo o apoio dedicado ao longo da conclusão deste trabalho.

## RESUMO

O presente trabalho propõe o desenvolvimento de uma aplicação *Web*, denominada *Health Clinic*, voltada para a otimização do acesso a serviços de saúde e aprimoramento da experiência do usuário no contexto dos estabelecimentos de saúde. O projeto busca enfrentar os desafios relacionados ao acesso da informação, oferecendo uma plataforma intuitiva e eficiente para a busca de clínicas e unidades de saúde, com base em suas especialidades médicas, oferecendo informações relativas ao atendimento e médicos, além da localização. O sistema foi desenvolvido com o apoio de tecnologias modernas, tais como *React.js*, *Node.js* e *MongoDB*, a aplicação prioriza a usabilidade e a praticidade, proporcionando uma interação amigável e acessível. Além disso, a plataforma visa contribuir para a organização e gestão eficaz das clínicas, promovendo um ambiente mais eficiente e colaborativo. Ao abordar a interseção entre tecnologia e saúde, o *Health Clinic* representa uma inovação significativa no setor, visando aprimorar o acesso aos cuidados médicos e proporcionar uma experiência mais positiva e eficiente para os usuários.

**Palavras-chave:** Aplicação *Web*. Clínicas médicas. Saúde.

## **ABSTRACT**

The present work proposes the development of an application Web, called Health Clinic, aimed at optimizing access to health services and improving the user experience in the context of clinics and health units. The project seeks to address challenges related to access to information, offering an intuitive and efficient platform for searching for clinics and health units, based on their medical specialties, offering information regarding care and doctors, in addition to location. The system was developed with the support of modern technologies, such as React.js, Node.js and MongoDB the application prioritizes usability and practicality, providing a friendly and accessible interaction. Furthermore, the platform aims to contribute to the organization and effective management of clinics, promoting a more efficient and collaborative environment. By addressing the intersection of technology and healthcare, Health Clinic represents a significant innovation in the industry, aiming to improve access to healthcare and provide a more positive and efficient experience for users.

**Keywords:** Web Application. Medical clinics. Health.



## LISTA DE FIGURAS

Figura 1 – Funcionamento do Padrão MVC . . . . .	19
Figura 2 – Bancos de dados <i>NoSQL</i> . . . . .	20
Figura 3 – Diagrama de Caso de Uso . . . . .	24
Figura 4 – Arquitetura do Sistema . . . . .	26
Figura 5 – Modelo Lógico do Banco de Dados . . . . .	28
Figura 6 – Tela de inicial da aplicação . . . . .	29
Figura 7 – Tela de login de usuários . . . . .	29
Figura 8 – Tela cadastro para clínica . . . . .	30
Figura 9 – Tela principal da clínica . . . . .	31
Figura 10 – Tela de adicionar médico a clínica . . . . .	32
Figura 11 – Tela criar agenda de funcionamento . . . . .	33
Figura 12 – Tela configurações . . . . .	34
Figura 13 – Tela principal paciente . . . . .	35

## LISTA DE QUADROS

Quadro 1 – Requisitos funcionais do sistema . . . . .	23
Quadro 2 – Caso de Uso - Cadastro . . . . .	24
Quadro 3 – Caso de Uso - Login . . . . .	24
Quadro 4 – Caso de Uso - Visualizar Clínicas . . . . .	25
Quadro 5 – Caso de Uso - Filtrar por Região . . . . .	25
Quadro 6 – Caso de Uso - Filtrar por Especialidade . . . . .	25
Quadro 7 – Caso de Uso - Editar Informações . . . . .	25
Quadro 8 – Caso de Uso - Criar Agenda de funcionamento . . . . .	25
Quadro 9 – Caso de Uso - Editar Agenda de funcionamento . . . . .	25
Quadro 10 – Caso de Uso - Adicionar Médico . . . . .	26
Quadro 11 – Comparação entre os sistemas . . . . .	37

## LISTA DE CÓDIGOS

Algoritmo 1 – Estrutura MongoDB - Orientado a documento . . . . .	21
---	----

## LISTA DE ABREVIATURAS E SIGLAS

API	<i>Application Programming Interface</i>
DOM	<i>Document Object Model</i>
HTTPS	<i>Hyper Text Transfer Protocol Secure</i>
JSON	<i>JavaScript Object Notation</i>
JWT	<i>JSON Web Token</i>
MVC	<i>Model-View-Controller</i>
NoSQL	<i>Not Only SQL</i>
OSS	<i>Open Source Software</i>
RF	Requisito Funcional
SQL	<i>Structured Query Language</i>
TCC	Trabalho de Conclusão do Curso
UI	<i>User Interface</i>
UX	<i>User Experience</i>
XML	<i>Extensible Markup Language</i>

# SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>10</b>
1.1	Motivação	11
1.2	Problemática	11
1.3	Objetivos	12
<b>1.3.1</b>	<b>Objetivo Geral</b>	<b>12</b>
<b>1.3.2</b>	<b>Objetivos Específicos</b>	<b>12</b>
1.4	Organização do documento	12
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	<b>14</b>
2.1	A tecnologia na saúde	14
2.2	Segurança em aplicações Web	14
2.3	React.js	15
<b>2.3.1</b>	<b>Biblioteca <i>Open Source</i></b>	<b>15</b>
<b>2.3.2</b>	<b>Componentização <i>React</i></b>	<b>16</b>
<b>2.3.3</b>	<b>Desenvolvimento <i>front-end</i></b>	<b>16</b>
2.4	Node.js	16
2.5	Arquitetura de Software	17
<b>2.5.1</b>	<b><i>Model-View-Controller</i></b>	<b>17</b>
2.6	NoSQL	19
<b>2.6.1</b>	<b>Banco de dados orientado a documentos</b>	<b>20</b>
<b>3</b>	<b>HEALTH CLINIC</b>	<b>22</b>
3.1	Levantamento de Requisitos	22
3.2	Análise dos Requisitos	22
<b>3.2.1</b>	<b><i>Stakeholders</i></b>	<b>22</b>
<b>3.2.2</b>	<b>Requisitos Funcionais</b>	<b>23</b>
<b>3.2.3</b>	<b>Diagrama de Casos de Uso</b>	<b>23</b>
3.3	Arquitetura do Sistema	26
3.4	Modelagem do Banco de Dados	27
3.5	Implementação	28
3.6	Trabalhos relacionados	35
<b>3.6.1</b>	<b>Processo de busca</b>	<b>36</b>
<b>3.6.2</b>	<b>Processo de seleção</b>	<b>36</b>
<b>3.6.3</b>	<b>Comparativo entre os sistemas</b>	<b>36</b>
<b>4</b>	<b>CONSIDERAÇÕES FINAIS</b>	<b>38</b>
4.1	Trabalhos Futuros	38
	<b>REFERÊNCIAS</b>	<b>40</b>

# 1 INTRODUÇÃO

Na contemporaneidade, a disseminação eficiente de informações sobre serviços de atendimento na área da saúde é um desafio crucial. Uma solução inovadora e altamente eficaz para superar essas barreiras é a implementação de uma aplicação web dedicada. Tal plataforma não apenas oferece um ponto centralizado de acesso às informações, mas também proporciona uma série de benefícios que impactam positivamente a relação entre os serviços de saúde e a comunidade (TELEMEDICINE. . . , 2015).

Uma das principais vantagens de uma aplicação web é o acesso facilitado às informações. Usuários podem rapidamente encontrar dados relevantes, como horários de funcionamento, localização de unidades de saúde, e detalhes sobre os serviços oferecidos. Essa facilidade de acesso é crucial, especialmente em momentos de emergência ou mudanças repentinas nas operações dos serviços de saúde (MEDWAY, 2022).

Diante disso, a tecnologia emerge como uma ferramenta potente capaz de desempenhar um papel significativo na aprimoração do acesso aos cuidados médicos. Em particular, a criação de aplicações *Web*, destinada a facilitar a busca de informações de clínicas de saúde e unidades de saúde, desponta como uma solução promissora. Essa abordagem não apenas atenua as barreiras geográficas, mas também contribui para otimizar a eficiência do sistema de saúde, conectando pacientes a serviços vitais de maneira mais rápida e eficaz.

Nesse contexto, este Trabalho de Conclusão de Curso (TCC) propõe o desenvolvimento de um sistema *Web*, denominado *Health Clinic*, com o objetivo de melhorar a comunicação e a troca de informação entre os estabelecimentos de saúde com o paciente, assim como proporcionar conforto e facilidades aos pacientes. Para a sua implementação, pretende-se utilizar em sua arquitetura as seguintes tecnologias: *Node.js*, *React.js* e *MongoDB*.

O sistema *Health Clinic* destaca-se pelo seu diferencial significativo ao priorizar a atenção à região do sertão paraibano. Com o objetivo primordial de proporcionar conforto aos usuários locais, o sistema oferece diversas funcionalidades, incluindo a localização de clínicas e unidades de saúde desejadas, visualização das agendas de funcionamento, além de informar sobre os médicos que atendem em determinada unidade, e sua respectiva agenda.

## 1.1 MOTIVAÇÃO

Na atualidade, a eficiente disseminação de informações sobre serviços de atendimento na área da saúde representa um desafio crucial. Reconhecendo a importância de superar tais barreiras, a implementação de uma aplicação web surge como uma solução inovadora e altamente eficaz. Essa plataforma não apenas oferece um ponto centralizado de acesso às informações, mas também introduz uma série de benefícios que impactam positivamente a relação entre os serviços de saúde e a comunidade. A facilidade de acesso a dados relevantes, como horários de funcionamento e localizações é fundamental, especialmente em momentos de emergência ou mudanças nas operações dos serviços de saúde.

A implementação da ferramenta proposta pretende fazer com que o paciente não precise sair da comodidade de sua casa para obter informações relativas ao atendimento médico prestado por clínicas e unidades de saúde. Quando doentes, este procedimento evita o contágio em uma possível sala de espera. Além disso, os profissionais de saúde e seus funcionários podem manter uma melhor organização, oferecendo aos seus clientes o fácil acesso às suas agendas médicas e informações.

## 1.2 PROBLEMÁTICA

As clínicas enfrentam desafios significativos na gestão do atendimento, especialmente em relação à sobrecarga de demanda e à ineficiência nos processos. O aumento contínuo no número de pacientes, aliado à complexidade das operações internas, resulta em longas filas de espera e atrasos nos horários de consulta. Esses problemas são agravados por sistemas de agendamento tradicionais que, muitas vezes, carecem de automação e integração eficazes. Consequentemente, a qualidade do atendimento personalizado é comprometida, uma vez que os profissionais de saúde têm menos tempo para se dedicar a cada paciente, dificultando a personalização dos cuidados de acordo com as necessidades individuais (BORDOLOI et al., 2019).

Além da sobrecarga interna, as clínicas também enfrentam desafios relacionados a barreiras geográficas, que limitam o acesso dos pacientes aos serviços de saúde. Em áreas remotas, a disponibilidade de clínicas é, muitas vezes, escassa, levando a dificuldades para os residentes locais em receber atendimento médico oportuno. Isso contribui para a falta de personalização nos cuidados, uma vez que os profissionais de saúde têm menos oportunidades de interação com os pacientes (ADLER-MILSTEIN et al., 2014).

A disseminação eficaz de informações sobre serviços de saúde é uma questão crítica na saúde pública. Diversos desafios dificultam a comunicação eficiente entre

os órgãos de saúde e a população. Fatores como a falta de acesso à informação, barreiras linguísticas, baixa alfabetização em saúde e desigualdades socioeconômicas contribuem para a dificuldade na divulgação de serviços disponíveis (LIMA et al., 2015).

A literatura destaca a importância de estratégias de comunicação culturalmente sensíveis, utilizando diversos canais de divulgação, como mídias sociais, rádio, televisão e panfletos, para atingir diferentes segmentos da sociedade. A colaboração entre profissionais de saúde, organizações não governamentais e líderes comunitários também é crucial para superar esses obstáculos.

## 1.3 OBJETIVOS

### 1.3.1 Objetivo Geral

O propósito principal deste Trabalho de Conclusão de Curso (TCC) é a criação de um sistema *Web* com o intuito de oferecer suporte a clínicas e unidades de saúde. Esse sistema tem como finalidade primordial auxiliar as equipes administrativas, permitindo-lhes compartilhar informações relevantes, como modalidades de atendimento, agenda médica e disponibilidade de horários. Além disso, busca-se aprimorar a visibilidade desses estabelecimentos de saúde por meio dessa plataforma. Em relação aos clientes, o que se visa é fornecer facilidades relativas à busca de informações, oferecendo o acesso a um atendimento ágil e de qualidade.

### 1.3.2 Objetivos Específicos

Com finalidade de atingir o objetivo geral, o trabalho proposto detém dos seguintes objetivos específicos:

- Analisar alguns sistemas semelhantes
- Adquirir conhecimentos sobre as tecnologias e ferramentas que possibilitam o desenvolvimento de aplicações *Web* de forma facilitada e organizada;
- Realizar o levantamento de requisitos da aplicação;
- Definir a estrutura do sistema com base nas tecnologias usadas;
- Desenvolver modelos de diagrama com o objetivo de documentar a proposta do sistema, facilitando sua compreensão e implementação.

## 1.4 ORGANIZAÇÃO DO DOCUMENTO

Este documento está organizado em quatro capítulos. O capítulo introdutório apresenta a motivação para a definição da proposta deste trabalho, seus objetivos, a



descrição de suas principais atividades e o seu cronograma de execução. O segundo capítulo fornece maiores informações sobre a tecnologia na saúde, conhecimentos/tecnologias a serem aplicados/as no trabalho, tais como *React.js*, *Node.js*, arquitetura de software, segurança em aplicações *Web* e NoSQL. No capítulo 3, são descritos detalhes do sistema *Web* proposto, o que inclui análise de requisitos, diagrama de casos de uso, modelo lógico do banco de dados e a implementação, este capítulo não apenas explora o sistema em profundidade, mas também vai além, contextualizando-o em relação aos trabalhos relacionados e realizando comparações pertinentes. Por fim, o último capítulo dispõe sobre as considerações finais e trabalhos futuros.

## 2 FUNDAMENTAÇÃO TEÓRICA

Neste tópico, são apresentados alguns pontos relevantes para o desenvolvimento do trabalho. Inicialmente, são descritas questões relacionadas com a área tema do sistema proposto: a tecnologia na saúde. Em seguida, são feitas breves descrições das tecnologias adotadas para o desenvolvimento da aplicação, assim como conhecimentos teóricos importantes que serão utilizados: arquitetura de software e banco de dados *NoSQL*.

### 2.1 A TECNOLOGIA NA SAÚDE

Um aspecto importante da tecnologia na saúde é sua capacidade de transformar a maneira como os profissionais de saúde prestam atendimento. Segundo o artigo "Tecnologia: definições e reflexões para a prática em saúde e enfermagem"(MARTINS; SASSO, 2008), a tecnologia tem possibilitado a evolução de serviços, comunidades, unidades e especialistas, dentro de ambientes que são referência na atenção aos clientes. Isso tem levado a mudanças nos valores, na política, na economia e nas relações humanas.

Além disso, a tecnologia na saúde desempenha um papel fundamental na promoção da equidade na saúde. Através da digitalização e da personalização do cuidado, a tecnologia permite que os profissionais de saúde forneçam cuidados mais precisos e adaptados às necessidades individuais dos pacientes. Isso é especialmente relevante para grupos vulneráveis, como idosos, pessoas com deficiências e populações rurais, que, frequentemente, enfrentam dificuldades para acessar serviços de saúde de alta qualidade. Além disso, a tecnologia também tem o potencial de reduzir as disparidades na saúde, pois possibilita a entrega mais eficiente e acessível de serviços de saúde, independentemente da localização geográfica ou do status socioeconômico do paciente.

### 2.2 SEGURANÇA EM APLICAÇÕES WEB

A segurança em aplicações web é um elemento crucial, destinado a assegurar a operação contínua e confiável dessas aplicações. Essa abordagem envolve uma variedade de medidas e técnicas projetadas para prevenir ameaças online e proteger os recursos envolvidos no desenvolvimento de aplicativos, incluindo servidores e processos associados à internet.

Uma prática de segurança fundamental é a implementação de um gerenciamento de senhas eficaz, incentivando o uso de senhas robustas que são alteradas

periodicamente. Além disso, a autenticação de dois fatores surge como uma opção viável para reforçar ainda mais a segurança do sistema (MDN, 2023).

O *JSON Web Token (JWT)*<sup>1</sup> é uma tecnologia de autenticação popular em aplicações web, permitindo que servidores verifiquem a identidade dos usuários sem armazenar sessões ou dados sensíveis. Composto por três partes (cabeçalho, conteúdo e assinatura), os tokens contêm informações do usuário, como ID. No entanto, apresentam desafios, como a falta de criptografia por padrão, tornando as informações vulneráveis. Para mitigar isso, é crucial transmitir os tokens por *HTTPS*, assegurando que não sejam interceptados durante a transmissão (JWT, 2023).

## 2.3 REACT.JS

O *React.js*<sup>2</sup> é uma biblioteca de *JavaScript* de código aberto desenvolvida pelo Facebook. Usado para construir interfaces de usuário (UI - *User Interface*) interativas e reativas para aplicativos *Web*, o *React* permite dividir a UI em componentes reutilizáveis e gerencia eficientemente as atualizações desses componentes quando ocorrem alterações nos dados. Esta tecnologia é amplamente utilizada para criar interfaces escaláveis de usuários e de alto desempenho (REACT, 2023). O uso do *React.js* proporciona uma maior modularidade no código desenvolvido, fazendo com que projetos complexos que sigam as boas práticas do *React.js* sejam de fácil manutenção, contribuindo para uma maior escalabilidade da aplicação.

### 2.3.1 Biblioteca *Open Source*

Uma aplicação *open source* se refere ao software que possui código projetado para ser acessado abertamente pelo público: todas as pessoas podem vê-lo, modificá-lo e distribuí-lo conforme suas necessidades (REDHAT, 2023). Muitos projetos *open source* são hospedados no *GitHub*<sup>3</sup>, ambiente em nuvem onde é possível acessar repositórios, compartilhar ou participar de projetos da comunidade aberta. Alguns dos projetos *open source* mais conhecidos são *Linux*, *Ansible*<sup>4</sup> e *Kubernetes*<sup>5</sup>.

O caso do *React.js* não é diferente. Em 2013, essa ferramenta foi disponibilizada em *JavaScript* na modalidade *open source*, desde então a sua popularidade tem apresentado um aumento gradativo, tendo em vista as aprovações no seu repositório no *GitHub*. Com suas diversas possibilidades para desenvolvimento *Web*, muitas empresas aderiram ao *React* em seus serviços e aplicações.

---

<sup>1</sup> <https://jwt.io/introduction/>

<sup>2</sup> <https://pt-br.react.dev/>

<sup>3</sup> <https://github.com/>

<sup>4</sup> <https://www.ansible.com/>

<sup>5</sup> <https://kubernetes.io/pt-br/>

### 2.3.2 Componentização *React*

Uma das principais características do *React.js* é seu modelo de programação baseado em componentes. Os componentes são blocos autônomos de código que representam partes específicas da interface do usuário. Eles podem ser combinados para formar interfaces complexas e reutilizáveis. Cada componente pode ter seu próprio estado interno e comportamento. Quando o estado de um componente é modificado, a interface da aplicação é alterada automaticamente para refletir o novo estado do componente (GACKENHEIMER, 2015).

No *React.js*, existem duas opções para criação de componentes: Componentes Funcionais e Componentes de Classe. No entanto, com o desenvolvimento dos *Hooks*, os Componentes de Classe estão sendo menos utilizados. Os *Hooks* permitem o uso de funções para alterar o estado de componentes, gerenciar ações durante o ciclo de vida de componentes, compartilhar contextos em toda a aplicação, entre outros. Outra característica importante dos *Hooks* é o da imutabilidade, em que ao invés de atualizar diretamente o valor das variáveis de estado da aplicação, a função *setState* do *Hook useState*, por exemplo, é utilizada para atribuir um novo estado.

### 2.3.3 Desenvolvimento *front-end*

O *React.js* é amplamente adotado para o desenvolvimento de *front-end*, ou seja, a parte da aplicação responsável pela interface do usuário. De acordo com Gackenheimer (2015), o *React.js* desafia as convenções estabelecidas e se destaca como uma alternativa que redefine as melhores práticas no desenvolvimento com *JavaScript*. Uma das razões pelas quais o *React.js* se destaca é por introduzir novos paradigmas e mudar o *status quo* em relação ao que é necessário para criar aplicativos *JavaScript* e interfaces de usuário escaláveis e sustentáveis.

Outro aspecto importante do *React.js* é a sua abordagem declarativa. Em vez de manipular diretamente o DOM (Modelo de Objeto de Documento), o *React.js* permite que os desenvolvedores descrevam como a interface deve ser exibida em diferentes estados da aplicação. Essa abordagem simplifica o processo de criação e atualização da interface, minimizando a possibilidade de erros. Ao escolher *React.js* como o *framework* de desenvolvimento *front-end* do sistema proposto neste trabalho, todos esses aspectos foram considerados.

## 2.4 NODE.JS

O *Node.js*<sup>6</sup>, criado por Ryan Dahl e outros colaboradores em 2009, introduziu uma abordagem inovadora para o desenvolvimento de aplicações, utilizando uma

<sup>6</sup> <https://nodejs.org/pt-br>

arquitetura não bloqueante. Essa abordagem permite que várias ações sejam executadas simultaneamente, tornando o *Node.js* adequado para lidar com aplicações de alto desempenho e com grande volume de dados. Uma das principais características do *Node.js* é a utilização da linguagem *JavaScript* como mecanismo de execução, o que proporciona um ambiente de tempo de execução eficiente do lado do servidor (BARSOTI; GIBERTONI, 2020).

De acordo com Barsoti e Gibertoni (2020), uma das vantagens do *Node.js* é a facilidade de construção de APIs (*Application Programming Interface*) e aplicações *NoSQL*. Sua abordagem orientada a eventos e sua forma eficiente de I/O (input/output) fazem com que ele seja leve e eficiente, permitindo um processamento rápido de solicitações e respostas.

Além disso, o ecossistema do *Node.js* é conhecido pela diversidade e abundância de bibliotecas e módulos disponíveis. Isso facilita o desenvolvimento de soluções, uma vez que os desenvolvedores podem aproveitar uma ampla gama de ferramentas e recursos prontamente disponíveis.

Essas características do *Node.js*, como sua arquitetura não bloqueante, a utilização do *JavaScript*, a facilidade na construção de APIs e aplicações *NoSQL*, bem como a disponibilidade de bibliotecas, são aspectos que foram levados em consideração ao escolher esse *framework* para o desenvolvimento do *back-end* do sistema proposto por este trabalho.

## 2.5 ARQUITETURA DE SOFTWARE

Garlan e Shaw (1993) tratam a arquitetura de um sistema como uma forma de solucionar problemas iminentes no desenvolvimento de um projeto, com questões estruturais que envolvem a organização total e estrutura de controle global; protocolos de comunicação, sincronização e acesso a dados; designação de funcionalidades a elementos de *design*; distribuição física; composição de elementos de *design*; escalação e desempenho; e seleção entre as alternativas de *design*. Com base nisso, este Trabalho de Conclusão de Curso, tem um desenvolvimento voltado para os protocolos de comunicação com a implementação do padrão *Model-View-Controller* (MVC), que possibilita a divisão do projeto em camadas muito bem definidas.

### 2.5.1 *Model-View-Controller*

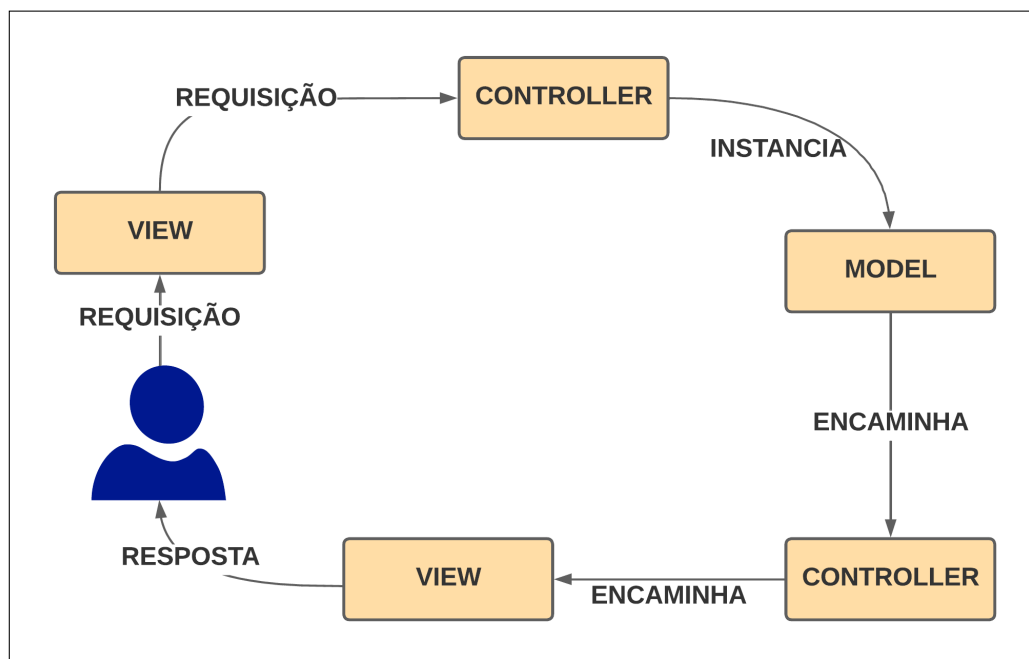
O *Model-View-Controller* é um padrão de arquitetura de software. Ele sugere uma maneira de tratar a divisão de responsabilidades em um projeto de software, principalmente em uma aplicação *Web*. O princípio básico é a divisão da aplicação em três camadas: a camada de interação do usuário, a camada de manipulação dos

dados e a camada de controle(LUCIANO; ALVES, 2017). Com isso, é possível separar o código relativo à interface do usuário das regras de negócio, o que sem dúvida traz muitas vantagens. A seguir, são listadas as responsabilidades de cada camada que compõe o MVC e, conseqüentemente, também estão presentes nos softwares que adotam nesta arquitetura.

- **Model:** camada responsável por armazenar e manipular os dados, além de fornecer métodos e funcionalidades para acessá-los e modificá-los. Ele é independente da interface do usuário e interage diretamente com o banco de dados ou outras fontes de dados para buscar, atualizar e persistir informações.
- **View:** camada responsável por apresentar os dados do modelo de forma visualmente agradável e interativa, permitindo que os usuários interajam com a aplicação. A *View* também pode enviar informações de volta para a camada *Controller* para que sejam processadas e atualizem o estado do modelo, iniciando assim um ciclo de interação entre as camadas.
- **Controller:** camada responsável por receber as solicitações do usuário, processar os dados necessários e tomar decisões sobre como a interface (*View*) deve ser atualizada e quais ações devem ser tomadas nos modelos (*Model*). O *Controller* atua como intermediário entre a *View* e o *Model*, garantindo a separação de responsabilidades e facilitando a interação entre os componentes do sistema.

Portanto, com o uso desse padrão de arquitetura, ganhamos algumas vantagens, como a redução de complexidade no código, aumento de produtividade, uniformidade na estrutura do software e reduz o tempo de desenvolvimento de um projeto (ACERVOLIMA, 2023). A Figura 1 apresenta um diagrama esquemático do funcionamento do padrão de arquitetura MVC.

Figura 1 – Funcionamento do Padrão MVC



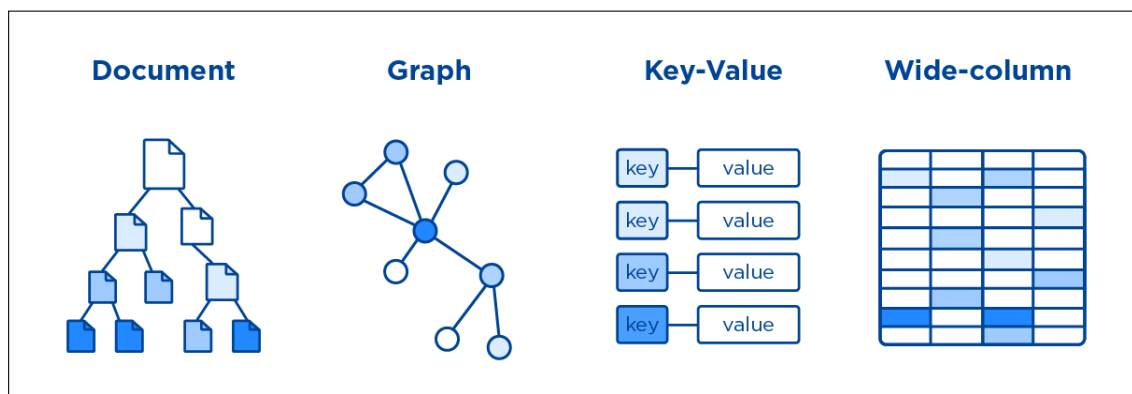
Fonte: Elaborado pelo autor

## 2.6 NOSQL

Com o crescimento exponencial da quantidade de dados e informações disponíveis na *Web*, especialmente devido ao surgimento das redes sociais, houve uma demanda crescente por sistemas capazes de lidar com tipos de dados complexos, semiestruturados e não estruturados. Esse cenário impulsionou significativamente as pesquisas sobre modelos de bancos de dados que podem lidar com esses desafios. Devido a esses motivos, surgiram os bancos de dados *NoSQL* (*Not Only SQL*) para abordar essa problemática, apresentando uma abordagem distinta para a persistência de dados, priorizando a disponibilidade, desempenho, escalabilidade e consistência dos dados, com fundamentos no paradigma *BASE* (PINTAR; POLETTTO, 2016).

Ao contrário dos modelos relacionais que possuem apenas um tipo de armazenamento, os bancos de dados *NoSQL* oferecem diversos tipos de armazenamento. Isso ocorre devido à necessidade de suportar qualquer tipo de dados, sejam eles complexos, semiestruturados ou não estruturados, resultando em vários modelos de dados. Esses modelos são categorizados em chave-valor, orientado a documentos, orientado a colunas e orientado a grafos (LÓSCIO et al., 2011). A Figura 2 faz referência a esses tipos de modelos de dados *NoSQL*.

Figura 2 – Bancos de dados NoSQL



Fonte: Abcloudz (2023)

### 2.6.1 Banco de dados orientado a documentos

Um banco de dados orientado a documentos pode ser definido como um modelo projetado para armazenar e consultar dados na forma de documentos *JSON* ou *XML*. Sua principal função é facilitar para os desenvolvedores o armazenamento e a consulta de dados, utilizando o mesmo formato de modelo de documento que eles usam no código ao desenvolver um aplicativo. Isso proporciona uma maior flexibilidade e agilidade no desenvolvimento, permitindo que os desenvolvedores trabalhem com estruturas de dados familiares e evitem a necessidade de mapear os objetos do aplicativo para um formato de tabela relacional (AWS, 2023).

A modelagem de banco de dados orientada a documentos utiliza pares de chave-valor para encapsular informações. Os documentos são as unidades básicas de armazenamento e não possuem uma estrutura rígida como nas tabelas do modelo relacional. Isso oferece vantagens significativas, como suporte a qualquer tipo de dados e maior flexibilidade na estrutura do documento. Essa flexibilidade permite a inclusão dinâmica de campos, sem alterar a estrutura da tabela. Essa abordagem é ideal para lidar com dados complexos, aninhados e heterogêneos, onde a estrutura dos dados pode variar e evoluir com o tempo. (JUNIOR et al., 2015)



**Algoritmo 1 – Estrutura MongoDB - Orientado a documento**

```
1 {  
2   "_id": ObjectId("615c0344c811b42dc809a8b9"),  
3   "nome": "Rafael",  
4   "email": "rafashpb@gmail.com"  
5   "cpf": "111.222.333-44"  
6   "idade": 21,  
7   "telefone": 8399999999,  
8   "genero": "Masc.",  
9   "nascimento": 25052002  
10 }
```

Fonte: Elaborado pelo autor

No Algoritmo 1, pode-se observar as informações de um usuário da aplicação proposta por este trabalho. Esses dados estão sendo armazenados dentro de um banco de dados de documentos *MongoDB*<sup>7</sup>, que é o adotado para a aplicação proposta. É possível ver também um identificador (chave) associado a ele (id). Dessa forma, as informações desse usuário podem ser consultadas com facilidade.

O *MongoDB* é amplamente reconhecido por sua capacidade de lidar com grandes volumes de dados, tornando-o uma escolha popular para aplicações que exigem escalabilidade e alta disponibilidade. Sua arquitetura flexível e distribuída permite que os dados sejam replicados e distribuídos em vários servidores, garantindo redundância e melhorando a tolerância a falhas (BANKER et al., 2016).

---

<sup>7</sup> <https://www.mongodb.com/>

## 3 HEALTH CLINIC

*Health Clinic* é uma aplicação *Web* desenvolvida com o objetivo específico de otimizar a busca por informações médicas relativas ao funcionamento e localização de clínicas e unidades de saúde. A aplicação foi concebida para fornecer suporte tanto aos pacientes quanto aos profissionais envolvidos na gestão dessas clínicas. Ela oferece uma maior visibilidade na região para os estabelecimentos de saúde, destacando-se pela ênfase na comodidade dos pacientes. O foco inicial do sistema está na facilitação do acesso aos serviços médicos, buscando, conseqüentemente, aprimorar a satisfação do cliente.

Nos tópicos abaixo, serão apresentadas atividades efetuadas durante a etapa de análise e projeto do sistema de gestão proposto. Inicialmente, é descrita a análise de requisitos, a arquitetura de software da aplicação e o modelo lógico do banco de dados projetado. Finalizando, alguns protótipos de telas desenvolvidos são ilustrados.

### 3.1 LEVANTAMENTO DE REQUISITOS

O levantamento de requisitos é uma etapa primordial que definirá quais funcionalidades deverão ser atendidas pela aplicação. Para isso, os requisitos foram estabelecidos com base na análise de sistemas existentes no mercado de propósitos semelhantes, o conhecimento adquirido sobre o funcionamento das clínicas médicas por meio de pesquisas e a experiência pessoal como paciente das mesmas.

### 3.2 ANÁLISE DOS REQUISITOS

Ao concluir o levantamento de requisitos, foram identificadas algumas funcionalidades que são essenciais para o cumprimento do objetivo do software proposto. A seguir, são definidas as partes interessadas do sistema, os requisitos funcionais e o diagrama de casos de uso.

#### 3.2.1 *Stakeholders*

- **Clínicas e Unidades básicas de saúde:** estes *stakeholders* têm a expectativa de que o sistema proposto otimize a visibilidade da instituição e melhore substancialmente a qualidade do atendimento prestado aos seus clientes, especialmente considerando a transmissão eficaz de informações para clientes de outras regiões e cidades menos favorecidas em termos de acesso a serviços de saúde. Dentro dessa categoria, estão incluídas tanto clínicas quanto as unidades básicas;

- **Pacientes:** Os pacientes desempenham um papel crucial como *stakeholders* no sistema proposto. Suas expectativas incluem a facilitação na busca por clínicas em determinadas regiões e especialidades, bem como a disponibilidade de suas informações para um atendimento prático e eficiente.

### 3.2.2 Requisitos Funcionais

No Quadro 1 são listados os requisitos funcionais do sistema, descrevendo as funcionalidades específicas que serão implementadas.

Quadro 1 – Requisitos funcionais do sistema

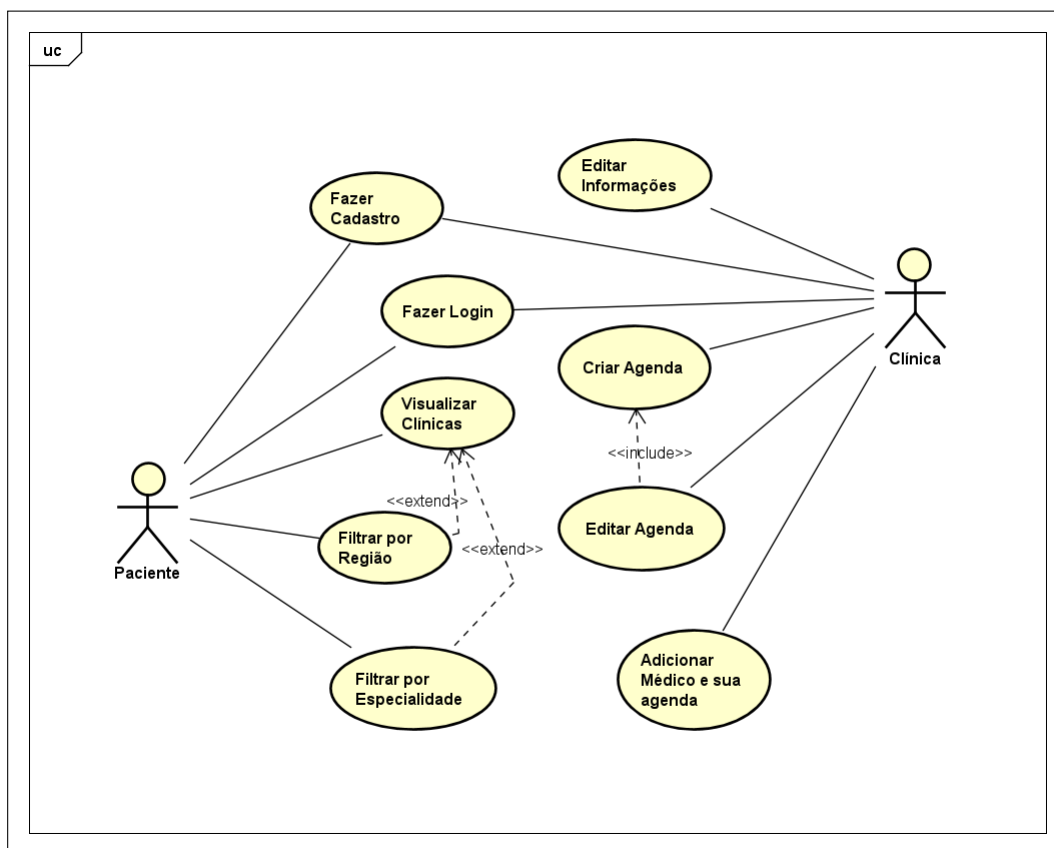
ID	DESCRIÇÃO
RF01	O sistema deve permitir que os usuários façam um cadastro na aplicação, solicitando a eles informações.
RF02	O sistema deve permitir o login dos usuários, solicitando algumas informações fornecidas na etapa de cadastro.
RF03	O sistema deve permitir que o <i>Paciente</i> faça a visualização das clínicas.
RF04	O sistema deve permitir que o <i>Paciente</i> filtre a pesquisa por clínicas de acordo com a especialidade e região.
RF05	O <i>Paciente</i> poderá visualizar a localização da clínica.
RF06	O sistema deve permitir que a <i>Clínica</i> crie uma agenda médica.
RF07	O sistema deve permitir que a <i>Clínica</i> faça alterações na agenda médica.
RF08	O sistema deve permitir que a <i>Clínica</i> adicione médicos que atendem em sua clínica.
RF09	O sistema deve possibilitar a visualização de informações da clínica, abrangendo horários de atendimento, médicos disponíveis e suas respectivas agendas.
RF10	O sistema deve permitir que os usuários deletem sua conta na aplicação.

Fonte: Elaborado pelo autor

### 3.2.3 Diagrama de Casos de Uso

Nesta seção, na Figura 3, é exibido o Diagrama de Casos de Uso elaborado para o sistema proposto neste trabalho. Bezerra (2007) afirma que este diagrama tem a função de representar graficamente as interações existentes entre os atores (usuários) e a aplicação a ser desenvolvida, demonstrando os principais casos de uso do sistema e como eles se relacionam.

Figura 3 – Diagrama de Caso de Uso



Fonte: Elaborado pelo autor

Na Figura 3, nota-se a presença de quatro atores: *Paciente*, *Clínica*. Cada um desses atores desempenha ações específicas e comuns no sistema, conforme definido no diagrama. A seguir, nos Quadros 3–11, são descritas as ações contidas no diagrama da Figura 3:

#### Quadro 2 – Caso de Uso - Cadastro

##### Caso de Uso 01 - Fazer cadastro

Todos os atores devem fornecer as informações necessárias para realizar seu cadastro no sistema.

Fonte: Elaborado pelo autor

#### Quadro 3 – Caso de Uso - Login

##### Caso de Uso 02 - Fazer login

Todos os atores acessam o sistema usando suas credenciais.

Fonte: Elaborado pelo autor

#### Quadro 4 – Caso de Uso - Visualizar Clínicas

##### Caso de Uso 03 - Visualizar Clínicas

O ator *Paciente*, seus generalizados têm acesso À visualização de uma lista de clínicas cadastradas na aplicação e às informações básicas.

Fonte: Elaborado pelo autor

#### Quadro 5 – Caso de Uso - Filtrar por Região

##### Caso de Uso 04 - Filtrar por Região

O ator *Paciente* e seus generalizados têm a opção de filtrar as clinicas por região desejada.

Fonte: Elaborado pelo autor

#### Quadro 6 – Caso de Uso - Filtrar por Especialidade

##### Caso de Uso 05 - Filtrar por Especialidade

O ator *Paciente* têm a opção de filtrar as clinicas pelas áreas de atuação desejada.

Fonte: Elaborado pelo autor

#### Quadro 7 – Caso de Uso - Editar Informações

##### Caso de Uso 06 - Editar Informações

O ator *Clínica* pode editar informações básicas do estabelecimento.

Fonte: Elaborado pelo autor

#### Quadro 8 – Caso de Uso - Criar Agenda de funcionamento

##### Caso de Uso 07 - Criar Agenda

O ator *Clínica* cadastra informações de disponibilidade na agenda de funcionamento, garantindo que os usuários possam visualizar a agenda atualizada.

Fonte: Elaborado pelo autor

#### Quadro 9 – Caso de Uso - Editar Agenda de funcionamento

##### Caso de Uso 08 - Editar Agenda

O ator *Clínica* pode editar a agenda de funcionamento para que os usuarios possam visualizar a agenda atualizada.

Fonte: Elaborado pelo autor

### Quadro 10 – Caso de Uso - Adicionar Médico

#### Caso de Uso 09 - Adicionar Médico

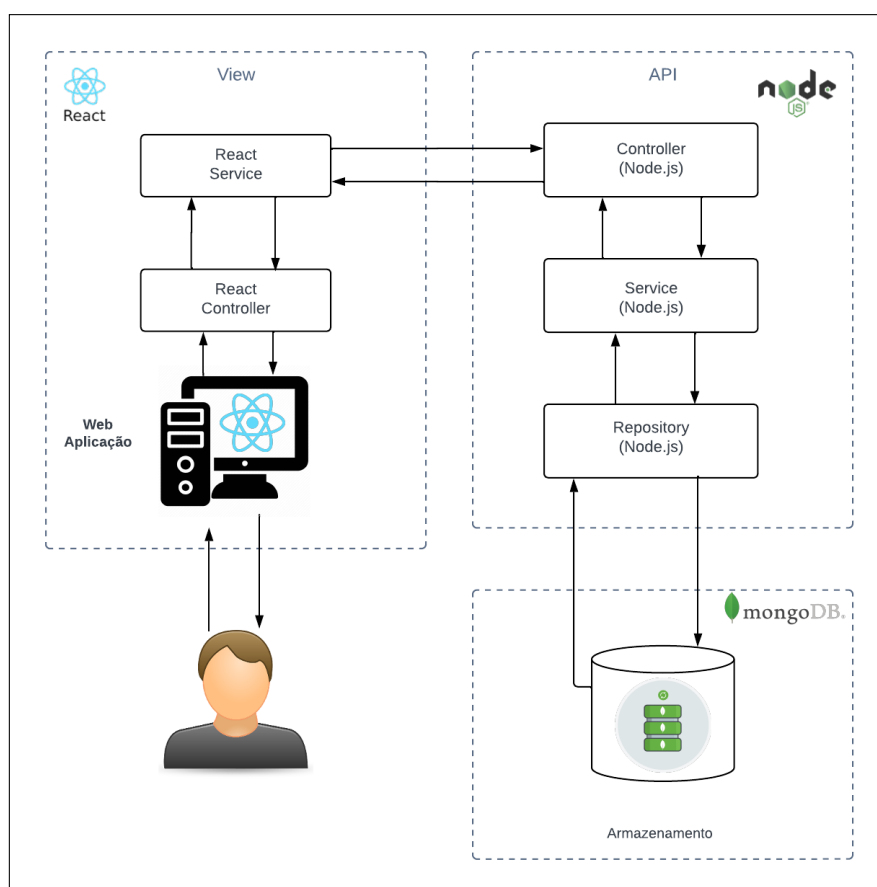
O ator *Clínica* têm a opção de adicionar um médico que atenderá em sua clínica, incluindo informações sobre o médico e os dias e horários de seu atendimento.

Fonte: Elaborado pelo autor

## 3.3 ARQUITETURA DO SISTEMA

O *Health Clinic* conta com uma arquitetura dividida em três camadas, baseada no padrão MVC: *View*, *Controller* e *Model*. Essa configuração fornece algumas vantagens no desenvolvimento da aplicação, tais como maior capacidade de manutenção e reutilização de código. A Figura 4 ilustra a arquitetura proposta para o sistema, onde podem ser vistas todas as camadas da aplicação e suas ligações:

Figura 4 – Arquitetura do Sistema



Fonte: Elaborado pelo autor

Para o sistema proposto, será desenvolvida uma interface dividida em três módulos. O primeiro é uma aplicação *Web* construída com *React*, que utiliza *JavaScript* para executar sua lógica e permitir uma interface de usuário responsiva. O segundo módulo, *React Controller*, será responsável por salvar, carregar e manipular os dados exibidos em tela. Já o terceiro, *React Service*, será responsável por manter uma conexão com a API e por salvar e recuperar informações.

A segunda camada é conhecida como API e deve ser criada utilizando o padrão *Controller-Service-Repository*. O *Controller* é responsável por gerenciar e estabelecer a comunicação com a camada de apresentação. O *Service*, por sua vez, processa e valida os dados antes de enviá-los para o terceiro módulo. Finalmente, o *Repository* cria uma comunicação com o banco de dados após receber os dados enviados.

Por fim, a camada de armazenamento deve ser responsável por armazenar qualquer informação da aplicação que for enviada para ela. Essa camada possui um sistema de gerenciamento de banco de dados, o *MongoDB*, que é utilizado para armazenar e gerenciar esses dados.

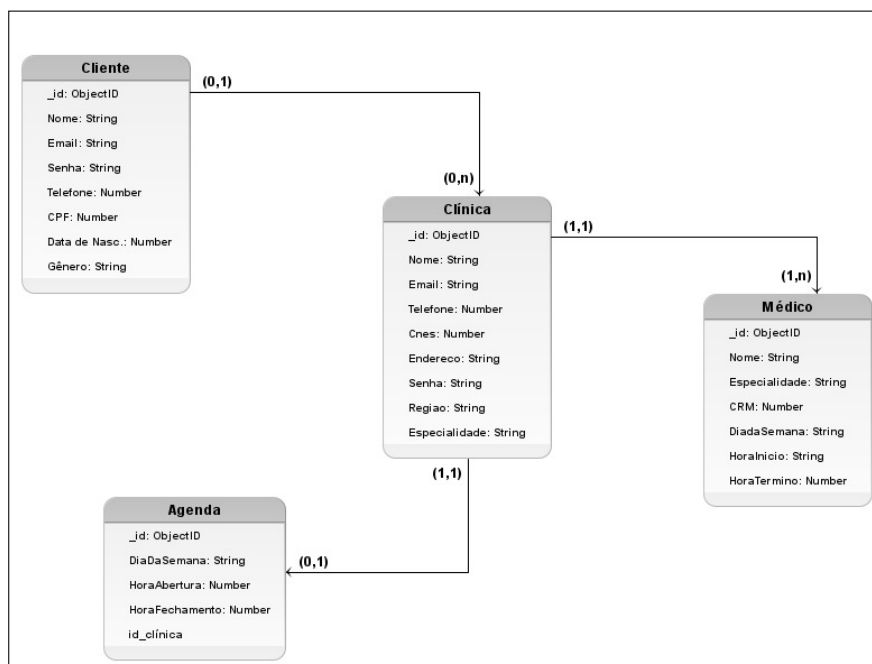
### 3.4 MODELAGEM DO BANCO DE DADOS

Segundo Elmasri e Navathe (2011), um banco de dados deve representar alguma característica do mundo real, sendo muitas vezes chamado de minimundo ou de universo de discurso (UoD - *Universe of Discourse*). Mouta (2020) descreve um minimundo como sendo uma porção da realidade, captada pelo analista, a qual a aplicação tem forte interesse em observar.

Dessa forma, Machado e Abreu (2004) definem que um banco de dados constitui uma coleção de fatos registrados que refletem o estado de certos aspectos de interesse do mundo real. A todo o momento o conteúdo do banco de dados representa uma visão instantânea do estado do mundo real. Cada mudança em algum item do banco de dados reflete uma mudança ocorrida na realidade.

Neste contexto, a representação lógica do banco de dados, mostrada na Figura 5, compõe o armazenamento dos dados da aplicação. Ela consiste na representação das principais tabelas a serem criadas na base de dados do sistema proposto, bem como os atributos que são gerados a partir dos relacionamentos entre as entidades.

Figura 5 – Modelo Lógico do Banco de Dados



Fonte: Elaborado pelo autor

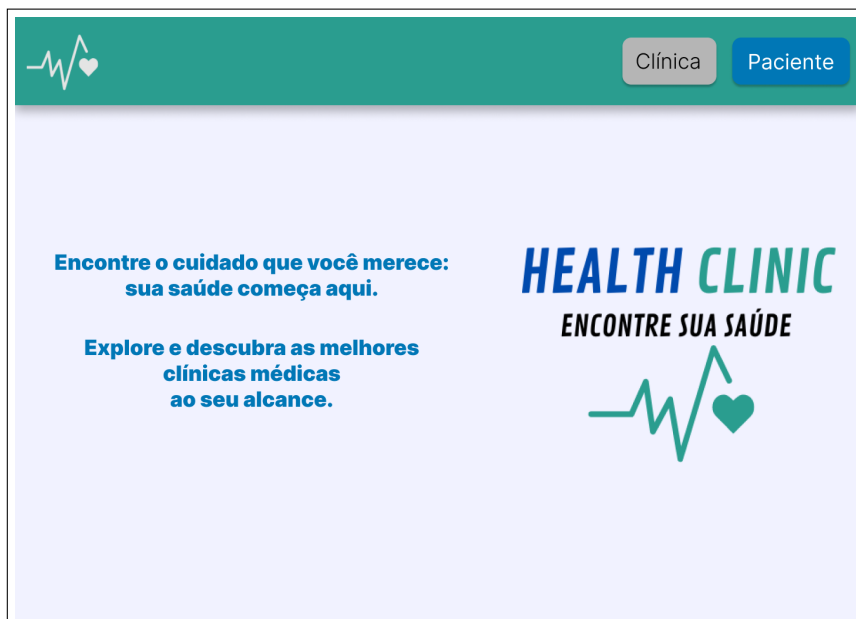
### 3.5 IMPLEMENTAÇÃO

Esta seção apresentará algumas telas do sistema e suas funcionalidades, que suprem os requisitos funcionais propostos no Quadro 1

inda na tela inicial da aplicação, apresentada na Figura 6, destaca-se o dashboard, onde são apresentadas as opções de usuários, sendo elas "Clínica" e "Paciente". Ao escolher uma dessas opções, o usuário será direcionado para a tela de login correspondente.



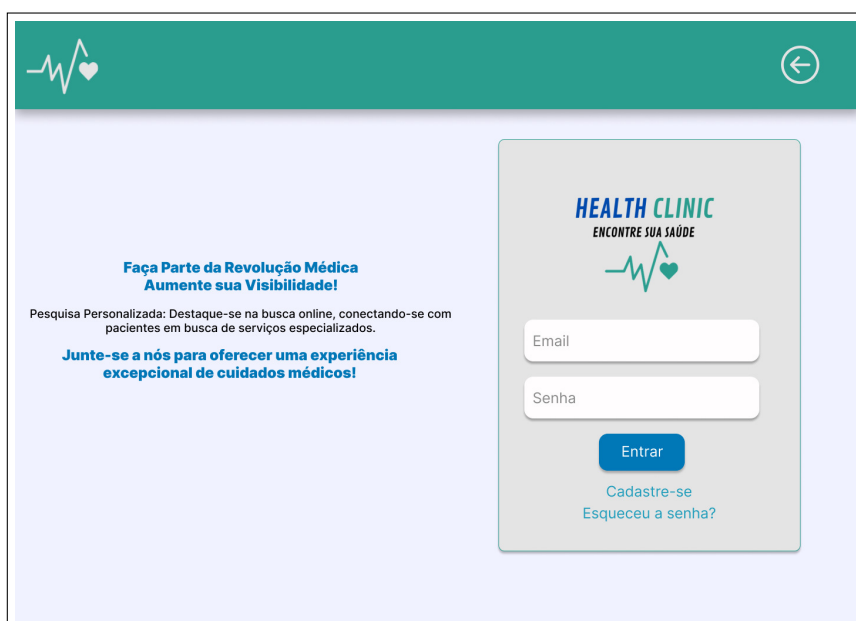
Figura 6 – Tela de inicial da aplicação



Fonte: Elaborado pelo autor

Seguindo o caminho da Clínica, ao clicar em 'Clínica', o usuário terá acesso à tela de login, apresentada na Figura 7. Nessa tela, terá a opção de efetuar o login, caso já possua um cadastro ativo, bem como as opções de se cadastrar e recuperar senha.

Figura 7 – Tela de login de usuários



Fonte: Elaborado pelo autor

Para um novo usuário que optar pela opção 'Cadastre-se', terá acesso à tela correspondente na Figura 8, onde poderá fornecer os dados necessários para ingressar no sistema. Os campos obrigatórios incluem: 'Nome', 'Email', 'Número', 'CNES' (um campo que assegura a participação apenas de clínicas com cadastro para funcionamento), 'Endereço', 'Senha', 'Região' e 'Especialidade'. Após fornecer esses dados e clicar no botão 'Finalizar', o usuário pode obter dois resultados possíveis. Em caso de sucesso, o cadastro será concluído, e o usuário será redirecionado para a tela de login. Em caso de erro, pode ocorrer de algum campo estar preenchido incorretamente ou já ser utilizado por outra clínica, como nos casos de 'Email' e 'Número'.

**Figura 8 – Tela cadastro para clínica**

A imagem mostra a interface de usuário para o cadastro de uma clínica. No topo, há uma barra verde com um ícone de coração e uma seta para trás. O formulário principal tem o título "FAÇA PARTE DESSA HISTÓRIA" e o subtítulo "CADASTRE-SE AUMENTE A VISIBILIDADE DE SUA CLÍNICA". Abaixo disso, há uma instrução: "Digite o nome de sua clínica". Os campos de entrada são: "Nome da clínica" (com o exemplo "clinica@gmail.com"), "Email" (com o exemplo "99 9 99999999"), "Número" (com o exemplo "99 9 99999999"), "Cnes da sua clínica" (com o exemplo "Número de 7, 10 ou 12 dígitos"), "Endereço" (com o exemplo "R. Dimas Andriola"), "Escolha uma senha" (com a instrução "A senha deve ter 8 caracteres, incluindo pelo menos um número"), "Em qual região a clínica se localiza?" (menu suspenso com "Selecione uma opção"), e "Qual especialidade da sua clínica?" (menu suspenso com "Selecione uma opção"). No final do formulário, há um botão azul com o texto "FINALIZAR".

Fonte: Elaborado pelo autor

Ao ser redirecionado para a tela de login, representada na Figura 7, o usuário, ao ingressar no sistema, terá acesso à tela reproduzida na Figura 9.

Figura 9 – Tela principal da clínica



Fonte: Elaborado pelo autor

Nessa tela, poderá visualizar suas próprias informações, e terá a opção de 'Adicionar Médico'. Esta opção disponibiliza um formulário de cadastro para inserir informações sobre médicos que realizam atendimentos e prestam serviços à sua clínica, Além de registrar os horários de atendimento, é possível incluir informações detalhadas sobre o médico, proporcionando uma visão abrangente do profissional para os usuários da plataforma. Essa funcionalidade serve como um meio de fornecer informações complementares para os usuários 'Pacientes'. Figura 10.

**Figura 10 – Tela de adicionar médico a clínica**

Adicione médicos que atendem em sua clínica!  
crie uma agenda médica

**Informações do Médico**

Digite o nome do Médico  
Nome do Médico

Qual especialidade do Médico?  
Selecione uma opção

Digite o CRM do Médico  
CRM

Preencha os dias nos quais o médico irá atender

Segunda-feira

Terça-feira

Quarta-feira

Quinta-feira

Sexta-feira

Sábado

Domingo

Selecione os horários de atendimento para Segunda-feira

Início:  
06:00

Termino:  
18:00

Adicionar Médico

Fonte: Elaborado pelo autor

Outra opção disponível na tela principal - Figura 9 - é a adição de uma agenda de funcionamento médica. Ao clicar nesta opção, o usuário será redirecionado para a tela de criação de agenda.

Na tela representada pela Figura 11, é possível visualizar as opções dos dias da semana representados por quadrados selecionáveis no formato de check-point. Ao selecionar a opção desejada, surgirá o campo para registrar o horário de abertura e encerramento do expediente na clínica. Em seguida, há um campo onde é possível digitar o e-mail para a confirmação da criação da agenda. Ao clicar em 'Salvar', a agenda será criada, e o usuário será redirecionado à tela principal.

Figura 11 – Tela criar agenda de funcionamento

Crie uma agenda de funcionamento para sua clínica!

Selecione os dias da semana

Segunda-feira  Terça-feira  Quarta-feira  Quinta-feira  Sexta-feira  Sábado  Domingo

Selecione os horários para Segunda-feira

Abre: 06:00 v

Fecha: 18:00 v

clinica@email.com

Salvar Agenda

Fonte: Elaborado pelo autor

Por fim, ao clicar no ícone de configuração localizado no cabeçalho da tela, representado pela engrenagem e ilustrado na Figura 12, surgirão as opções de editar dados, remover médicos e excluir conta. Essas opções proporcionam ao usuário a capacidade de modificar informações pessoais, gerenciar médicos associados à clínica e, se necessário, excluir a conta.

Figura 12 – Tela configurações



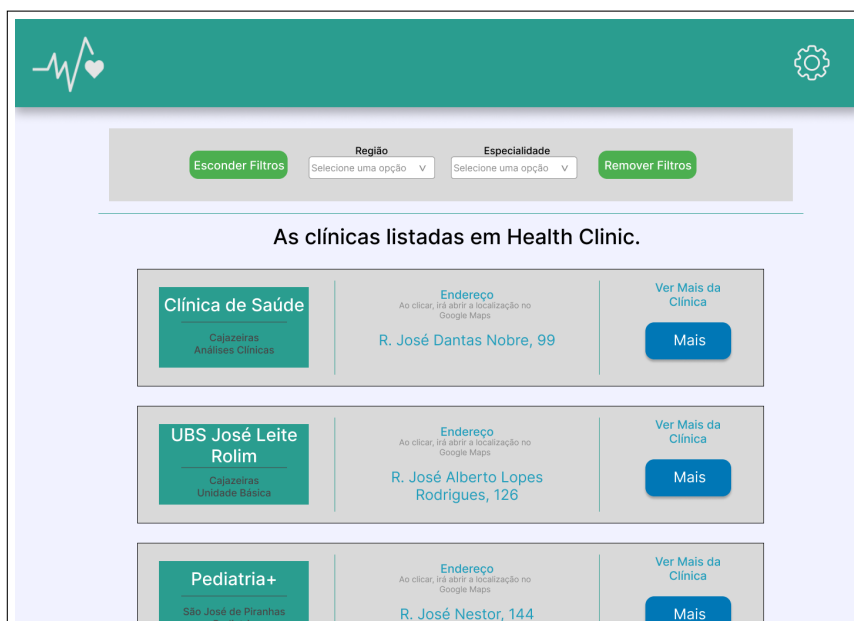
Fonte: Elaborado pelo autor

Iniciando um caminho convencional para o usuário 'Paciente', ao escolher a opção correspondente na tela inicial, seguirá o mesmo processo descrito anteriormente, apresentando as mesmas opções, e diferenciando apenas na tela de cadastro. Nessa tela, será disponibilizado um formulário destinado a pessoas físicas.

Ao ingressar na aplicação, a tela principal do usuário 'Paciente', mostrada na Figura 13, exibe uma lista de cards das clínicas cadastradas na aplicação. Esses cards incluem informações como Nome, Região, Especialidade e Endereço, sendo possível visualizar a localização no Google Maps ao clicar. Além disso, há a opção 'Ver Mais', representada como um botão que, ao ser clicada, redireciona para uma página com informações mais detalhadas sobre a clínica desejada.

Na parte superior da tela, abaixo do cabeçalho, estão disponíveis opções de filtros. Esses filtros são baseados nas escolhas do usuário, sendo determinadas por Região e Especialidade. Ao selecionar uma ou ambas as opções, a lista de cards é atualizada instantaneamente, mostrando apenas os resultados da escolha definida. Esses filtros servem para o benefício do usuário, proporcionando praticidade na busca de forma intuitiva e eficiente. Vale ressaltar que as clínicas cadastradas na aplicação são meramente ilustrativas, e não fazem parte de uma parceria com clínicas reais.

**Figura 13 – Tela principal paciente**



Fonte: Elaborado pelo autor

Ao optar pela opção 'Ver Mais', o usuário terá acesso a uma tela semelhante à Figura 9, porém apenas informativa. Essa tela conterá os dados básicos da clínica e informações complementares, como os médicos associados e a opção de visualizar os horários de atendimento, além da opção de visualizar a agenda de funcionamento da própria clínica.

Em conclusão, é evidente que a evolução da tecnologia desempenha um papel fundamental na melhoria da experiência do usuário. No mundo digital atual, a acessibilidade e a praticidade tornaram-se aspectos cruciais para garantir a satisfação dos usuários. Nesse contexto, destacamos a adaptabilidade das telas, tornando-as responsivas e totalmente acessíveis por meio de navegadores móveis. A capacidade de acessar informações, realizar tarefas e interagir com a aplicação de forma intuitiva, por meio de dispositivos móveis, é um significativo avanço. Esse avanço não apenas facilita a vida cotidiana, mas também amplia as possibilidades de conectividade, garantindo que as informações estejam ao alcance de todos, a qualquer momento e em qualquer lugar.

### 3.6 TRABALHOS RELACIONADOS

Para a execução deste projeto, foi indispensável realizar a busca e seleção criteriosa de materiais que estivessem alinhados com os objetivos relacionados ao

tema abordado. A seguir, apresentam-se os tópicos essenciais que fundamentam a validação do processo de filtragem para a pesquisa do conteúdo desenvolvido.

### 3.6.1 Processo de busca

Para a elaboração deste trabalho, escolheu-se o tema amplo "Inovações tecnológicas na saúde com foco na otimização do acesso e aprimoramento da comunicação". Para conduzir a pesquisa, a plataforma utilizada foi o Google, aliada à busca específica em sites governamentais das cidades situadas na região do Sertão Paraibano.

### 3.6.2 Processo de seleção

A seleção dos trabalhos se deu de acordo com os seguintes critérios adotados: (1) O sistema encontrado é sobre auxílio de busca voltada para estabelecimentos de saúde? (2) O sistema encontrado é baseado na democratização do acesso à informação? (3) O sistema encontrado provê uma comunicação assertiva com base no público alvo?

Ao final, com base nos filtros de seleção, temos 2 sistemas existentes:

- Meu SUS Digital<sup>1</sup>: é um programa desenvolvido pelo Governo Federal do Brasil que prevê a informatização e integração dos dados de saúde dos cidadãos entre estabelecimentos de saúde e os órgãos de gestão em saúde dos entes federativos.
- Página da Secretaria de Saúde de Cajazeiras-PB<sup>2</sup>: página *Web* com intuito de exibir informações sobre o órgão pertencente à gestão pública municipal, que tem como missão: Garantir o acesso dos serviços de atendimento aos usuários do SUS; planejar e executar o desenvolvimento de projetos e programas de Saúde; fiscalizar o controle de condições sanitárias, higiênicas e de medicamentos; promover campanhas educativas de proteção a população, no que se refere a epidemias e outras doenças; desenvolver estudos, avaliações e o devido controle sobre o funcionamento dos serviços de atendimento.

### 3.6.3 Comparativo entre os sistemas

Em um mundo cada vez mais impulsionado pela tecnologia e inovação, a escolha da ferramenta certa é primordial para atender as necessidades específicas de indivíduos e organizações; um quadro de comparação de funcionalidades emerge

<sup>1</sup> <https://meusdigital.saude.gov.br/>

<sup>2</sup> <https://www.cajazeiras.pb.gov.br/secretaria.php?sec=22/>



como uma ferramenta valiosa para avaliar e contrastar diferentes soluções disponíveis no mercado.

O quadro 11 se propõe a oferecer uma visão abrangente das funcionalidades oferecidas pelas opções tratadas na seção anterior e do sistema proposto neste TCC, permitindo uma análise sistemática e eficaz.

**Quadro 11 – Comparação entre os sistemas**

<b>Funcionalidades</b>	<b>Meu SUS</b>	<b>SEC Saúde</b>	<b>Health Clinic</b>
Disponibilidade de Informações	Sim	Sim	Sim
Lista de Estabelecimentos	Sim	Sim	Sim
Opção de Filtro	Sim	Não	Sim
Geolocalização	Sim	Sim	Não
Horários de Funcionamento	Sim	Sim	Sim
Agenda de Médicos	Não	Não	Sim
Edição de Dados	Não	Não	Sim

Fonte: Elaborado pelo autor

## 4 CONSIDERAÇÕES FINAIS

A criação e implementação da aplicação *Web Health Clinic* representam um avanço significativo no campo da gestão de serviços de saúde, com foco na facilitação do acesso e na melhoria da experiência do usuário. Ao longo deste trabalho, exploramos a importância de soluções tecnológicas inovadoras para superar os desafios enfrentados pelos usuários na busca por informações e serviços médicos em regiões onde ocorre a hierarquia urbana. Em tais cenários, cidades menos favorecidas, em termos de serviços de saúde e disseminação de informações, muitas vezes buscam assistência em áreas mais desenvolvidas.

A plataforma desenvolvida visa atender às crescentes demandas por praticidade e eficiência no setor de saúde, proporcionando aos usuários uma interface intuitiva para a busca de clínicas, especialidades médicas e unidades públicas de saúde. Ao priorizar a usabilidade, a aplicação visa eliminar barreiras geográficas e promover um acesso mais rápido e descomplicado aos serviços de saúde. Ademais, reconhecemos que a implementação bem-sucedida do Health Clinic não apenas beneficia os usuários finais, mas também as clínicas de saúde, ao proporcionar um alcance de visibilidade maior e a atenção do público alvo.

Portanto, concluímos que a aplicação web Health Clinic apresenta-se como uma solução promissora para aprimorar o acesso aos serviços de saúde, destacando-se pelo seu enfoque na experiência do usuário e na eficiência operacional das clínicas. Este trabalho não apenas abre portas para futuras pesquisas e aprimoramentos tecnológicos, mas também reforça a importância de investir em inovações que promovam uma saúde mais acessível e centrada no usuário.

### 4.1 TRABALHOS FUTUROS

A busca por aprimoramento contínuo nos leva à implantação de um aplicativo mobile, visando aumentar a eficiência e agilidade. Dessa forma, oferecemos a possibilidade de acessar a plataforma tanto via web quanto por dispositivos móveis. Essa abordagem reflete o aspecto evolutivo, considerando que o uso de smartphones está mais acessível a uma grande parcela da população.

Além disso, uma das inovações fundamentais que cabe nessa aplicação consiste na implementação de uma agenda única, unificando os horários dos médicos disponíveis. Essa funcionalidade visa otimizar um futuro sistema de marcação de consultas, proporcionando aos usuários uma visão abrangente dos horários disponíveis de diversos profissionais de saúde em um único local. A agenda única simplifica o processo de agendamento, permitindo que os pacientes encontrem facilmente horários

compatíveis com suas necessidades e preferências. Busca-se explorar, de maneira efetiva, a integração da aplicação proposta em ações públicas, com o principal propósito de aprimorar consideravelmente o acesso à saúde. A inclusão desta aplicação em programas governamentais de saúde pública poderia ampliar sua abrangência para comunidades mais extensas. A colaboração entre a aplicação e as iniciativas públicas oferece uma perspectiva promissora para transformar, de maneira positiva, o cenário da saúde, proporcionando um acesso mais equitativo e eficiente aos serviços médicos essenciais.

## REFERÊNCIAS

- ABCLOUDZ. **NoSQL Technologies**. 2023. <https://abcloudz.com/services/nosql-databases/>. Acessado em 6 de julho de 2023.
- ACERVOLIMA. **BENEFÍCIO DE USAR MVC**. 2023. <<https://acervolima.com/beneficio-de-usar-mvc/>>. Acessado em 12 de maio de 2023.
- ADLER-MILSTEIN, J.; KVEDAR, J.; BATES, D. W. Telehealth among us hospitals: several factors, including state reimbursement and licensure policies, influence adoption. **Health Affairs**, v. 33, n. 2, p. 207–215, 2014.
- AWS. **O que é um banco de dados de documentos?** 2023. <<https://aws.amazon.com/pt/nosql/document/>>. Acessado em 19 de maio de 2023.
- BANKER, K.; GARRETT, D.; BAKKUM, P.; VERCH, S. **MongoDB in action: covers MongoDB version 3.0**. [S.l.]: Simon and Schuster, 2016.
- BARSOTI, N.; GIBERTONI, D. Impacto que o sequelize traz para o desenvolvimento de uma api construída em node. js com express. js. **Revista Interface Tecnológica**, v. 17, n. 2, p. 231–243, 2020.
- BEZERRA, E. **Princípios de Análise e Projeto de Sistemas com UML**. [S.l.]: Elsevier Rio de Janeiro, 2007. v. 2.
- BORDOLOI, S.; FITZSIMMONS, J. A.; FITZSIMMONS, M. J. **Service management: operations, strategy, information technology**. [S.l.]: McGraw-Hill, 2019.
- ELMASRI, R.; NAVATHE, S. B. **Sistemas de Banco de Dados**. 6. ed. São Paulo: Person, 2011.
- GACKENHEIMER, C. **Introduction to React**. [S.l.]: Apress, 2015.
- GARLAN, D.; SHAW, M. An introduction to software architecture. In: **Advances in software engineering and knowledge engineering**. [S.l.]: World Scientific, 1993. p. 1–39.
- JUNIOR, R. G. R. et al. Migração de banco de dados relacional para orientado a documentos estudo de caso do projeto siade. Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte, 2015.
- JWT. **Introduction to JSON Web Tokens**. [S.l.]: Auth0, 2023. <<https://jwt.io/introduction/>>.
- LIMA, K. W. S. d.; ANTUNES, J. L. F.; SILVA, Z. P. d. Percepção dos gestores sobre o uso de indicadores nos serviços de saúde. **Saúde e Sociedade**, SciELO Brasil, v. 24, p. 61–71, 2015.
- LÓSCIO, B. F.; OLIVEIRA, H. d.; PONTES, J. d. S. Nosql no desenvolvimento de aplicações web colaborativas. **VIII Simpósio Brasileiro de Sistemas Colaborativos**, sn, v. 10, n. 1, p. 11, 2011.
- LUCIANO, J.; ALVES, W. J. B. Padrão de arquitetura mvc: Model-view-controller. **EPEQ Fafibe**, v. 1, n. 3a, p. 102–107, 2017.

MACHADO, F. N. R.; ABREU, M. P. **Projeto de banco de dados: uma visão prática**. 10. ed. São Paulo: Érica, 2004.

MARTINS, C. R.; SASSO, G. T. M. D. **Tecnologia: definições e reflexões para a prática em saúde e enfermagem**. [S.l.]: SciELO Brasil, 2008. 11–12 p.

MDN. **Web security basics**. [S.l.]: Mozilla Corporation, 2023. <[https://developer.mozilla.org/pt-BR/docs/Learn/Server-side/First\\_steps/Website\\_security](https://developer.mozilla.org/pt-BR/docs/Learn/Server-side/First_steps/Website_security)>.

MEDWAY. **Tecnologia na saúde**. 2022. Disponível em: <<https://vx.med.br/tecnologia-na-saude/>>.


MOUTA, A. E. B. **Modelagem de banco de dados sem segredos**. 2020. <<https://alberteije.com/blog/artigos/modelagem-bd-01.php>>. Acessado em 07 de julho de 2023.

PINTAR, P. H. R.; POLETTO, A. S. R. de S. Um estudo exploratório acerca de banco de dados nosql comparado aos bancos de dados relacionais. **FÓRUM CIENTÍFICO DA FEMANAIS**, p. 15, 2016.

REACT. **Introdução ao React**. 2023. <https://pt-br.react.dev/>. Acessado em 30 de maio de 2023.

REDHAT. **what is open source?** 2023. <https://www.redhat.com/pt-br/topics/open-source/what-is-open-source>. Acessado em 2 de junho de 2023.

TELEMEDICINE and Telehealth. **MedlinePlus Medical Encyclopedia**, 2015. Disponível em: <<https://medlineplus.gov/telemedicine.html>>.

	<b>INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA</b>
	Campus Cajazeiras - Código INEP: 25008978
	Rua José Antônio da Silva, 300, Jardim Oásis, CEP 58.900-000, Cajazeiras (PB)
	CNPJ: 10.783.898/0005-07 - Telefone: (83) 3532-4100

## Documento Digitalizado Ostensivo (Público)

### TCC

<b>Assunto:</b>	TCC
<b>Assinado por:</b>	Rafael Barroso
<b>Tipo do Documento:</b>	Anexo
<b>Situação:</b>	Finalizado
<b>Nível de Acesso:</b>	Ostensivo (Público)
<b>Tipo do Conferência:</b>	Cópia Simples

Documento assinado eletronicamente por:

- **Rafael Barroso de Santana, ALUNO (202012010006) DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS - CAJAZEIRAS**, em 08/03/2024 10:40:35.

Este documento foi armazenado no SUAP em 08/03/2024. Para comprovar sua integridade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifpb.edu.br/verificar-documento-externo/> e forneça os dados abaixo:

Código Verificador: 1108279  
Código de Autenticação: 0ddf1c8a2c

