

**CENTRO UNIVERSITÁRIO PARA O DESENVOLVIMENTO DO ALTO VALE DO  
ITAJAÍ - UNIDAVI**

**LUCAS VALENTE**

**CHATBOT ONLINE PARA ENCONTRAR PROFISSIONAIS DA SAÚDE**

**RIO DO SUL  
2022**

**CENTRO UNIVERSITÁRIO PARA O DESENVOLVIMENTO DO ALTO VALE DO  
ITAJAÍ - UNIDAVI**

**LUCAS VALENTE**

**CHATBOT ONLINE PARA ENCONTRAR PROFISSIONAIS DA SAÚDE**

Trabalho de Conclusão de Curso a ser apresentado ao curso de Sistemas da Informação, da Área das Ciências Naturais, da Computação e das Engenharias, do Centro Universitário para o Desenvolvimento do Alto Vale do Itajaí, como condição parcial para a obtenção do grau de Bacharel em Sistemas de Informação.

Prof. Orientador: Cleber Nardelli

**RIO DO SUL  
2022**

**CENTRO UNIVERSITÁRIO PARA O DESENVOLVIMENTO DO ALTO VALE DO  
ITAJAÍ - UNIDAVI**

**LUCAS VALENTE**

**CHATBOT ONLINE PARA ENCONTRAR PROFISSIONAIS DA SAÚDE**

Trabalho de Conclusão de Curso a ser apresentado ao curso de Sistemas da Informação, da Área das Ciências Naturais, da Computação e das Engenharias, do Centro Universitário para o Desenvolvimento do Alto Vale do Itajaí- UNIDAVI, a ser apreciado pela Banca Examinadora, formada por:

---

Professor Orientador: Cleber Nardelli

Banca Examinadora:

---

Prof. M.e Jullian Hermann Creutzberg

---

Prof. M.e Fernando Andrade Bastos

Rio do Sul, 29 de novembro de 2022.

Ao meus pais por todo o apoio.

## **AGRADECIMENTOS**

Aos meus pais, amigos e familiares que sempre me deram apoio durante todo o período de formação acadêmica, meus amigos Cauê, Danielle e Diogo e também a minha namorada Eduarda, que sempre me incentivaram a continuar estudando e evoluindo.

Ao professor e orientador deste trabalho, Cleber Nardelli, no qual aceitou a compartilhar seu tempo e conhecimento para ajudar a criar este projeto de forma excelente e pensar no futuro dele.

A minha mãe, um agradecimento muito especial por ter sempre me apoiado quando precisei, e não ter deixado eu desistir nos momentos mais difíceis. Não só isso, mas também sempre ter feito muitos esforços para que eu pudesse estudar e chegar aonde eu cheguei. Obrigado mãe, amo você.

## RESUMO

O setor da saúde vem se mostrando cada vez mais importante para o bem-estar da sociedade, principalmente com o aumento populacional, e com o aumento tecnológico exponencial cada vez mais é necessário a criação de novas tecnologias para auxiliar toda a população. Este trabalho tem como objetivo desenvolver um protótipo de aplicação web com o objetivo de disponibilizar um *chatbot* online para que as pessoas busquem profissionais da saúde através da conversa com o robô, e além disso, permitir aos profissionais que façam seu cadastro na plataforma para aumentar seu alcance profissional através do *chatbot*. Para realizar a pesquisa, a metodologia utilizada foi a de uma pesquisa aplicada e descritiva e para que os objetivos sejam alcançados foi realizada uma revisão da literatura sobre as principais tecnologias e linguagens de programação utilizadas no desenvolvimento, além das ferramentas e outras metodologias que envolvem o desenvolvimento de software. Também foi realizada uma pesquisa por uma ou mais ferramentas que oferecem um serviço semelhante ao protótipo proposto, visando a construção da seção do estado da arte. O capítulo de desenvolvimento detalha toda a análise que contém o levantamento de requisitos e regras de negócio, e também demonstra as funcionalidades do sistema através dos fluxogramas, diagrama caso de uso e entidade relacionamento. Por fim na seção de implementação estão apresentados todos os detalhes sobre o desenvolvimento e funcionamento do protótipo, juntamente com um diagrama de distribuição da aplicação. O protótipo proposto permite com que através dele seja possível como profissional da saúde se cadastrar e fazer parte da plataforma para que outras pessoas o encontrem, ampliando seu alcance. Em contrapartida, outras pessoas poderão conversar através do *chatbot*, como sendo um assistente para procurar por profissionais em regiões e áreas desejadas.

**Palavras-Chave:** Chatbot, assistente, saúde.

## ABSTRACT

The health sector has proved to be increasingly important for the well-being of society, especially with population growth, and with the exponential technological increase, it becomes increasingly necessary to create new technologies to help the entire population. This work aims to develop a prototype of a web application with the objective of making an online chatbot available for people to look for health professionals through conversation with the robot, and in addition, to allow professionals to register on the platform to increase their reach. professional. through the chatbot. To carry out the research, the methodology used was that of an applied and descriptive research and for the objectives to be achieved, a literature review was carried out on the main technologies and programming languages used in the development, in addition to the tools and other methodologies that involve software development. A search was also carried out for one or more tools that offer a service similar to the proposed prototype, aiming at building the state-of-the-art section. The development chapter details all the analysis that contains the survey of requirements and business rules, and also demonstrates the system's functionalities through flowcharts, use case diagrams and relationships between entities. Finally, in the implementation section, all details about the development and operation of the prototype are presented, along with an application distribution diagram. The proposed prototype allows the health professional to register and be part of the platform so that other people can find it, expanding its reach. On the other hand, other people will be able to chat through the chatbot, as an aid to search for professionals in the desired regions and areas.

**Keywords:** Chatbot, assistant, health.

## LISTA DE FIGURAS

Figura 1 - Sintaxe geral de um elemento HTML.....	23
Figura 2 - Estrutura básica de um elemento HTML.....	24
Figura 3 - Sintaxe de uma regra CSS .....	24
Figura 4 - Ilustração do Iubem para pacientes no dispositivo iOS.....	30
Figura 5 - Ilustração do Iubem para profissionais no dispositivo iOS .....	31
Figura 6 - Protótipo da tela de consulta de locais de atuação.....	33
Figura 7 - Protótipo da tela de manutenção das especialidades .....	34
Figura 8 - Tela inicial com o chatbot aberto e o robô iniciando a conversa.....	34
Figura 9 - Diagrama de caso de uso do médico e do usuário anônimo. ....	38
Figura 10 - Fluxograma do processamento de linguagem natural no robô .....	39
Figura 11 - Fluxograma para solicitar um profissional da saúde .....	40
Figura 12 - Diagrama de modelo entidade e relacionamento.....	41
Figura 13 - Criação de rotas com Express.js .....	42
Figura 14 - Diagrama de Distribuição da Aplicação .....	44
Figura 15 - Tela inicial do sistema (RF02).....	45
Figura 16 - Cadastro de usuário no Sistema (RF07, RF10, RF12).....	45
Figura 17 - Tela de login (RF06).....	46
Figura 18 - Formulário de cadastro de dados pessoais no primeiro acesso do usuário (RF01).....	47
Figura 19 - Tela de Especialidades do Profissional (RF11, RF14) .....	47
Figura 20 - Formulário de Cadastro de um Local de Atuação (RF18).....	48
Figura 21 - Consulta de Locais de Atuação (RF13) .....	48
Figura 22 - Conversa de robô x usuário (RF05, RF09, RF16) .....	49

## LISTA DE QUADROS

Quadro 1 - Pontos importantes para a definição da qualidade de software.....	16
Quadro 2 - Vantagens do Node.js.....	20
Quadro 3 - Tipos de banco de dados .....	22
Quadro 4 - Aplicabilidade da NLP .....	26
Quadro 5 - Princípios da LGDP .....	27
Quadro 6 - Regras de Negócio .....	35
Quadro 7 - Requisitos não funcionais .....	36
Quadro 8 - Requisitos funcionais .....	36

## **LISTA DE ABREVIATURAS E SIGLAS**

API	Application programming interface
BFF	Back-end for front-end
FTP	File transfer protocol
HTTP	Hypertext transfer protocol
IOT	Internet of things
NLP	Natural language processing
PLN	Processamento de linguagem natural
SGBD	Sistema de gerenciamento de banco de dados
SMTP	Simple mail transfer protocol
TCP	Transmission control protocol

## SUMÁRIO

<b>1. INTRODUÇÃO</b> .....	<b>11</b>
1.1 PROBLEMA DE PESQUISA .....	12
1.2 OBJETIVOS .....	12
<b>1.2.1 Geral</b> .....	<b>12</b>
<b>1.2.2 Específicos</b> .....	<b>12</b>
1.3 JUSTIFICATIVA .....	12
<b>2. REFERENCIAL TEÓRICO</b> .....	<b>14</b>
2.1 ENGENHARIA DE SOFTWARE .....	14
<b>2.1.1 Engenharia de requisitos</b> .....	<b>14</b>
<b>2.1.2 Qualidade de software</b> .....	<b>15</b>
2.2 JAVASCRIPT .....	16
2.3 TYPESCRIPT.....	17
2.4 BACK-END DA APLICAÇÃO .....	17
<b>2.4.1 Protocolo HTTP</b> .....	<b>18</b>
<b>2.4.2 Node.js</b> .....	<b>19</b>
<b>2.4.3 Express.js</b> .....	<b>20</b>
<b>2.4.4 API</b> .....	<b>20</b>
<b>2.4.4 Rest</b> .....	<b>21</b>
<b>2.4.5 Banco de dados</b> .....	<b>21</b>
<b>2.4.5.1 Structured Query Language</b> .....	<b>22</b>
<b>2.4.5.2 PostgreSQL</b> .....	<b>22</b>
2.5 FRONT-END DA APLICAÇÃO .....	23
<b>2.5.1 HTML</b> .....	<b>23</b>
<b>2.5.2 CSS</b> .....	<b>24</b>
<b>2.5.3 React.js</b> .....	<b>25</b>
2.6 NLP.....	25
<b>2.6.1 Node NLP</b> .....	<b>26</b>
2.7 LGPD.....	26
<b>2.7.1 Dados anonimizados</b> .....	<b>27</b>
<b>3. METODOLOGIA DA PESQUISA</b> .....	<b>29</b>
3.1 ESTADO DA ARTE .....	29
<b>3.1.1 Iubem</b> .....	<b>29</b>
<b>4. CHATBOT ONLINE PARA ENCONTRAR PROFISSIONAIS DA SAÚDE</b> .....	<b>32</b>
4.1. ANÁLISE .....	32

<b>4.1.1. Visão Geral.....</b>	<b>32</b>
<b>4.1.2 Comparação do Protótipo com o Estado da Arte.....</b>	<b>32</b>
<b>4.1.3. Prototipação .....</b>	<b>33</b>
4.2. REGRAS DE NEGÓCIO .....	35
4.3 REQUISITOS .....	35
4.4 DIAGRAMAS .....	37
4.5 IMPLEMENTAÇÃO .....	41
4.6 UTILIZAÇÃO E FUNCIONAMENTO.....	42
<b>4.6.1 Estrutura de distribuição da aplicação.....</b>	<b>42</b>
<b>4.6.2 Funcionamento .....</b>	<b>44</b>
<b>5. CONSIDERAÇÕES FINAIS.....</b>	<b>50</b>
5.1 RECOMENDAÇÕES DE TRABALHOS FUTUROS .....	51

## 1. INTRODUÇÃO

No século XXI a importância dos cuidados com a saúde está ficando cada vez mais evidente, principalmente após a pandemia do COVID-19 que iniciou no ano de 2020. Contudo com o avanço tecnológico as tarefas diárias se tornaram cada vez mais práticas, e isso inclui o setor da saúde. Devido ao período de isolamento social, muitas clínicas, médicos e outros profissionais da saúde tiveram que mudar seu atendimento para o formato online, através de videochamadas. Mesmo após o período de isolamento muitos dos profissionais continuam adotando esse método de trabalho visto que isso permite um alcance maior de pacientes. Entretanto apesar desta mudança, é visível que para encontrar profissionais ainda é muito complicado visto que muitos não tem um site especializado ou algum meio de contato fácil pela internet.

A proposta do *chatbot* tem como objetivo disponibilizar um assistente e auxiliar as pessoas a encontrarem profissionais da saúde de uma forma rápida e centralizada, podendo ser utilizado por toda a população em geral. Os usuários podem solicitar a ele que informe um ou mais profissionais de uma determinada especialidade em uma determinada região, e ele pode visualizar as informações do profissional selecionado para entrar em contato e agendar uma consulta ou contratar seus serviços. O intuito do protótipo é permitir que as pessoas encontrem profissionais da saúde perto deles, ou da região que eles selecionarem, de uma forma rápida e fácil.

Além do capítulo introdutório, esse trabalho possui uma revisão da literatura onde serão abordadas todas as tecnologias usadas no desenvolvimento do protótipo, fornecendo conceito sobre todas elas. No terceiro capítulo temos a metodologia da pesquisa que explica detalhadamente o processo metodológico deste trabalho. No quarto capítulo é apresentado o desenvolvimento da aplicação junto com toda a parte de análise do projeto. Por fim, no último capítulo são listadas as considerações finais.

## 1.1 PROBLEMA DE PESQUISA

Como ajudar as pessoas a encontrarem profissionais da saúde especializados em determinadas áreas, tanto pelo celular quanto pelo computador, sendo que geralmente eles não ficam indexados no Google?

## 1.2 OBJETIVOS

Nesta seção serão detalhados os objetivos gerais e específicos do protótipo.

### 1.2.1 Geral

- Desenvolver o protótipo de um *chatbot* para encontrar profissionais da saúde.

### 1.2.2 Específicos

- Levantar requisitos para desenvolver o *chatbot*.
- Definir o método de configuração da PLN (Processamento de Linguagem Natural) do robô.
- Modelar a estrutura de dados para armazenar no SGBD PostgreSQL.
- Definir a interface de operação do usuário por meio de prototipação.
- Implementar o protótipo com base na especificação de requisitos.
- Pesquisar sobre uma ou mais ferramentas já existentes no mercado que possuem uma finalidade semelhante.

## 1.3 JUSTIFICATIVA

A ajuda médica sempre foi necessária dentro das sociedades, seja para tratar alguma ferida ou até mesmo realizar um parto. Com o passar do tempo, à medida que a população e as tecnologias aumentaram, paralelamente a área da saúde também evoluiu, e com isso o número de profissionais na saúde cresceu naturalmente.

Nos dias atuais existem diversos cursos e faculdades disponíveis no mercado que possibilitam cada vez mais que as pessoas se profissionalizem em algum dos vários segmentos da saúde, como por exemplo enfermagem, psicologia, psiquiatria, cirurgiões, etc.

Entretanto apesar do grande número de serviços disponíveis, a busca por um profissional da saúde continua difícil, pois é necessário encontrar um que seja do agrado do paciente, seja no atendimento, na localização, nos convênios aceitos e até mesmo nas formas de pagamentos. Devido a essa dificuldade, as avaliações médicas acabam acontecendo de maneira informal, ou

seja, onde a própria sociedade troca informação entre si (também conhecimento popularmente pelo “boca-a-boca”), ou até mesmo através de pedidos por referências médicas em algum posto de atendimento gratuito da cidade.

No ano de 2021, o autor Marchi realizou uma pesquisa científica para desenvolver uma aplicação de catálogo médico. O objetivo da aplicação é centralizar os médicos de uma cidade, para que dessa forma os pacientes possam buscar todos os profissionais da cidade desejada através de uma consulta no catálogo médico do aplicativo e assim inclusive, poderem realizar agendamentos de consultas médicas. Além disso, os médicos podem gerenciar seus agendamentos e consultas através da aplicação, centralizando seu serviço em um só lugar (MARCHI, 2021).

Tendo o problema em vista, foi proposta a criação de uma aplicação para centralizar a busca de profissionais da saúde, visando não só solucionar o problema das pessoas, mas também ser um complemento ao trabalho científico de Marchi, pois o trabalho proposto tem como objetivo ter uma abrangência maior de regiões e de segmentos da saúde que serão aceitos na plataforma, além de utilizar de uma forma conversacional para realizar as buscas.

Com isso, a aplicação irá possibilitar ao profissional cadastre-se na plataforma e informe suas especialidades e locais de atendimento. Desta forma, os pacientes poderão buscar por profissionais através do *chatbot*, informando apenas a região (estado e cidade) e a especialidade ou segmento que deseja buscar.

## 2. REFERENCIAL TEÓRICO

Nesse capítulo serão abordados assuntos referentes as ferramentas de desenvolvimento, de modelagem e as tecnologias utilizadas no processo de desenvolvimento da aplicação.

### 2.1 ENGENHARIA DE SOFTWARE

Pressman e Maxim (2021, p. 5) afirmam que:

Software consiste em: (1) instruções (programas de computador) que, quando executadas, fornecem características, funções e desempenho desejados; (2) estruturas de dados que possibilitam aos programas manipular informações adequadamente; e (3) informação descritiva, tanto na forma impressa quanto na virtual, descrevendo a operação e o uso dos programas.

De acordo com Sommerville (2011), os sistemas de software são abstratos e intangíveis, de forma que não são algo materializado e que possa ser tocado fisicamente por alguém. Com isso um sistema pode ter um limite extremamente grande ou até não ter limite para expandir seu potencial, o que pode ser bom por um lado, contudo em contrapartida também pode se tornar muito complexo, com um entendimento difícil e caro para dar manutenção.

O objetivo da engenharia de software é apoiar o desenvolvimento profissional de um software, indo além da programação. Com isso, ela oferece técnicas que apoiam a especificação, projeto e a evolução de programas, que são coisas que geralmente são ignoradas ou não recebem a devida importância quando se trata de um desenvolvimento de software pessoal (SOMMERVILLE, 2011).

#### 2.1.1 Engenharia de requisitos

Segundo Sommerville (2011), os requisitos descrevem o que o sistema deve fazer, quais serviços irá oferecer e que tipo de restrições apresentará em seu funcionamento. O processo de levantamento de requisitos, ou engenharia de requisitos, deve ser feita junto ao cliente, para poder entender qual a sua necessidade e assim poder alcançar uma finalidade determinada.

Pressman e Maxim (2021, p. 104) afirma que:

“Uma parte importante do levantamento é entender as metas de negócios [...]. Uma meta é um objetivo de longo prazo que um sistema ou produto deve alcançar. As metas podem tratar de preocupações funcionais ou não funcionais (p. ex., confiabilidade, segurança, usabilidade) [...]”.

De acordo com Sommerville (2011), um dos problemas que surgem durante a etapa de levantamento de requisitos, é a não organização dos requisitos conforme seus tipos, que são separados entre requisitos de usuário e requisitos de sistema.

Sommerville (2011, p. 58) afirma que “Requisitos de usuário são declarações, em uma linguagem natural com diagramas, de quais serviços o sistema deverá fornecer a seus usuários e as restrições com as quais este deve operar.”

O mesmo autor também afirma que os requisitos de sistema estão mais ligados a detalhes de funções, serviços e restrições operacionais do sistema. Esse documento pode fazer parte do contrato entre o comprador e o vendedor do sistema, e deverá definir exatamente o que deve ser implementado (SOMMERVILE, 2011).

De acordo com Sommerville (2011), os requisitos de software geralmente são classificados como funcionais e não funcionais. Os requisitos funcionais estão muito ligados com o tipo de software e com os possíveis usuários. Geralmente descrevem funções que o sistema deve fazer. Podem ser escritos tanto de uma forma mais detalhada e específica, quanto de forma mais abstrata para que seja compreendido pelo usuário do sistema, dependendo do requisito. Em contrapartida os requisitos não funcionais descrevem funcionalidades emergentes, geralmente coisas como confiabilidade, tempo de resposta, etc.

### **2.1.2 Qualidade de software**

Pressman e Maxim (2021, p. 312) afirmam que: “[...] a qualidade de software pode ser definida como: uma gestão de qualidade efetiva aplicada de modo a criar um produto útil que forneça valor mensurável para aqueles que o produzem e para aqueles que o utilizam.”.

Para que a qualidade do software seja atingida o sistema além de fornecer todas as funcionalidades previstas de forma correta, também precisa agradar ao usuário, de forma que ele fique satisfeito e consiga atingir seus objetivos com o sistema pois, caso o contrário, não atinge a qualidade desejada (SOMMERVILE, 2011).

De acordo com Pressman e Maxim (2021) é importante enfatizar três pontos importantes para a definição da qualidade de software, sendo essa estrutura representada pelo Quadro 1.

**Quadro 1 - Pontos importantes para a definição da qualidade de software**

<b>Gestão de qualidade efetiva</b>	Uma gestão de qualidade efetiva estabelece a infraestrutura que dá suporte a qualquer tentativa de construir um produto de software de alta qualidade. Os aspectos administrativos do processo criam mecanismos de controle e equilíbrio de poderes que ajudam a evitar o caos no projeto – um fator-chave para uma qualidade inadequada. As práticas de engenharia de software permitem ao desenvolvedor analisar o problema e elaborar uma solução consistente – aspectos críticos na construção de software de alta qualidade. Finalmente, as atividades de apoio, como o gerenciamento de mudanças e as revisões técnicas, têm muito a ver com a qualidade, assim como qualquer outra parte da prática de engenharia de software.
<b>Produto útil</b>	Um produto útil fornece o conteúdo, as funções e os recursos que o usuário deseja; além disso, e não menos importante, deve fornecer confiabilidade e isenção de erros. Um produto útil sempre satisfaz às exigências definidas explicitamente pelos envolvidos. Além disso, ele satisfaz a um conjunto de requisitos implícitos (p. ex., facilidade de uso) que se espera de todo software de alta qualidade.
<b>Agregar valor</b>	Ao agregar valor tanto para o fabricante quanto para o usuário de um produto de software, um software de alta qualidade gera benefícios para a empresa de software e para a comunidade de usuários. A empresa fabricante do software ganha valor agregado pelo fato de um software de alta qualidade exigir menos manutenção, menos correções de erros e menos suporte ao cliente. Isso permite que os engenheiros de software despendam mais tempo criando aplicações novas e menos tempo em manutenções. A comunidade de usuários ganha um valor agregado, pois a aplicação fornece a capacidade de agilizar algum processo de negócio. O resultado final é: (1) maior receita gerada pelo produto de software; (2) maior rentabilidade quando uma aplicação suporta um processo de negócio; e/ou (3) maior disponibilidade de informações cruciais para o negócio.

Fonte: Adaptado de Pressman e Maxim (2021).

## 2.2 JAVASCRIPT

Segundo Flanagan (2014, p. 19):

Javascript foi criada na Netscape na fase inicial da Web e, tecnicamente, “Javascript” é marca registrada, licenciada pela Sun Microsystems (agora Oracle), usada para descrever a implementação da linguagem pelo Netscape (agora Mozilla). A Netscape enviou a linguagem para a ECMA – European Computer Manufacturer’s Association – para padronização e, devido a questões relacionadas à marca registrada, a versão padronizada manteve o nome estranho “ECMAScript”. Pelos mesmos motivos ligados à marca registrada, a versão da Microsoft da linguagem é formalmente conhecida como “JScript”. Na prática, quase todo mundo chama a linguagem de Javascript.

De acordo com Oliveira e Zanetti (2020), o Javascript foi criado com o objetivo de oferecer uma interatividade melhor entre as páginas, se baseando e tendo uma sintaxe semelhante à da linguagem Java, sendo também uma linguagem de programação orientada a objetos, porém que pode ser interpretada e executada pelos navegadores web.

O Javascript tem como função controlar os comportamentos da página, ou seja, isso inclui ações como realizar a validação de formulários, alterar textos ou estilos, ocultar e mostrar objetos e até mesmo executar pequenas operações e manipulações junto ao navegador (MILETTO; BERTAGNOLLI, 2014).

### 2.3 TYPESCRIPT

De acordo com Lopes (2017), a linguagem Typescript é relativamente nova no mercado, tendo surgido em outubro de 2012. Essa linguagem é considerada um superste do Javascript, e ela vai existir somente em tempo de desenvolvimento pois antes de executar o código, ele precisa ser *transpilado*. Dessa forma em tempo de execução, a linguagem que será interpretada é o Javascript.

Lopes (2017, p.13) afirma que “Pode-se misturar o Javascript e Typescript durante a escrita do código, porém, o código depurado será sempre Javascript. Existem algumas ferramentas que reverterem o código para ser depurado em Typescript [...]”.

O mesmo autor também declara que o Typescript é uma linguagem com tipagem estática e com um grande suporte para o desenvolvimento orientado a objetos, com tudo não há uma obrigatoriedade de realizar o desenvolvimento somente desta maneira. De forma geral o Typescript em si é apenas uma ferramenta para melhorar o desenvolvimento, possibilitando o uso de checagem de tipos e sintaxes mais modernas, enquanto ao mesmo tempo também permite o uso da sintaxe original do Javascript e seus recursos, sempre que for necessário (LOPES, 2017).

### 2.4 BACK-END DA APLICAÇÃO

Freitas (2021, p. 206) afirma que: “O *back end* é a parte responsável por lidar com os servidores, banco de dados, web services e, em específico, com as regras de negócios das aplicações”.

Segundo Souto (2019) o *back-end* faz a ponte entre os dados que o usuário envia do navegador, por exemplo, até ao banco de dados, fazendo toda a aplicação das regras de negócio e validações necessárias antes disso.

### 2.4.1 Protocolo HTTP

Comer (2015, p. 706) afirma que “O protocolo usado para a comunicação entre um navegador e um servidor web ou entre máquinas intermediárias e servidores web é conhecido como Hypertext Transfer Protocol (HTTP).”.

O HTTP funciona como uma combinação entre dois protocolos: o FTP, pois realiza a transferência de arquivos e também utiliza serviços TCP, sendo bem mais simples pois usa somente uma conexão TCP, sem a necessidade de uma conexão de controle separada, fazendo com que somente os dados sejam enviados durante a conexão do cliente com o servidor. E o outro protocolo é o SMTP, tendo como principal semelhança o formato das mensagens que são enviadas. Contudo os dados enviados pelo protocolo HTTP são lidos e interpretados pelo servidor HTTP, diferente do SMTP onde eles são destinados a serem lidos por humanos. Além disso a forma ao qual o protocolo HTTP transmite os dados e dá a resposta ao usuário também é diferente (FOROUZAN, 2010).

Forouzan (2010, p. 628) afirma que “[...] embora o HTTP use os serviços do TCP, ele próprio é um protocolo sem estado. O cliente inicializa a transação enviando uma mensagem de pedido. O servidor responde enviando uma resposta.”.

Segundo o mesmo autor, ao realizar uma requisição existe um formato de envio e de resposta. No envio é necessário definir um método HTTP, a URL ao qual será requisitada, os cabeçalhos e as vezes um corpo (FOROUZAN, 2010).

O protocolo HTTP define um conjunto de métodos de requisição, de forma que cada um indica uma ação que deverá ser executada e tem uma semântica diferente implementada (MDN CONTRIBUTORS, 2021b). Dentre os nove métodos, nesse projeto serão explorados apenas quatro deles:

- Get – solicita dados ao servidor, ou seja, esse método deve apenas retornar dados.
- Post – envia dados para o servidor e geralmente executa alguma mudança no estado do recurso ou efeitos colaterais no servidor.
- Put – substitui as representações do recurso de destino pela carga de dados enviada na requisição.
- Delete – remove um recurso específico.

Segundo Saudate (2014), no final de toda a requisição o servidor retorna um código de status, que são separados em cinco categorias:

- 1xx – Informativos;
- 2xx – Códigos de sucesso;
- 3xx – Códigos de redirecionamento;
- 4xx – Erros causados pelo cliente;
- 5xx – Erros originados no servidor.

## 2.4.2 Node.js

Oliveira e Zanetti (2021, p. 8) afirmam que “O Node.js é um ambiente de servidor de código aberto que possibilita a execução de aplicações escritas em Javascript e pode atuar, em aplicações para a Internet, como uma linguagem de programação server-side.”.

Para Oliveira e Zanetti (2021), realizar o desenvolvimento com Node.js, permite com que toda a versatilidade do Javascript possa ser utilizada no lado servidor, possibilitando a criação de aplicações robustas e escaláveis, junto com todos os recursos que a linguagem original oferece.

De acordo com os mesmos autores, o Node.js está alinhado com as principais modernidades do desenvolvimento de software, como por exemplo, a utilização de um padrão de arquitetura de entrada e saída assíncrona e, a conexão com banco de dados não relacionais (como o MongoDB). Além disso, também oferece algumas vantagens em relação aos seus principais concorrentes quando se refere a leveza, flexibilidade, suporte e produtividade (OLIVEIRA; ZANETTI, 2021).

Melo (2021, n.p.) afirma que:

A principal característica do Node.js é sua execução ser single-thread, ou seja, os recursos computacionais são alocados apenas uma vez pelo tempo que a aplicação estiver sendo executada. Aplicações multi-thread, como as criadas com PHP, por exemplo, requerem a criação de uma thread a cada nova requisição, e ela não é executada enquanto a anterior não for finalizada.

Nas palavras de Melo (2021) esse thread único é chamado de *Event Loop*. Ela tem como característica o fato de eliminar a necessidade de filas de processamento, pois trata todas as requisições como eventos, que são executados de forma assíncrona e não-bloqueável, tornando forma as aplicações ficam mais eficientes e responsivas. O Node.js foi baseado no V8 *engine* criado pelo Google para o navegador Chrome, e é um ambiente de execução em uma máquina virtual própria para interpretar e executar os scripts de forma autônoma.

Segundo Duarte (2021) o Node.js oferece algumas vantagens, que estão estruturadas no Quadro 2.

**Quadro 2 - Vantagens do Node.js**

<b>Usa Javascript</b>	Javascript é uma das linguagens de programação mais populares do mundo, e o fato do Node.js utilizar essa linguagem é muito bom pois possibilita que mais pessoas possam utiliza-lo, e até mesmo abre mais oportunidades de emprego.
<b>Permite Javascript full-stack</b>	Uma grande reclamação de muitos programadores web é ter de trabalhar com linguagens diferentes no <i>front-end</i> e no <i>back-end</i> . Node.js resolve isso ao permitir que você trabalhe com JS em ambos, principalmente se usar frameworks modernos de front como React.
<b>É muito leve e multiplataforma</b>	Isso permite que você consiga rodar seus projetos em servidores abertos e com o SO que quiser, diminuindo bastante seu custo de hardware e software.

Fonte: Adaptado de Duarte (2021).

Para Duarte (2021) o Node.js pode ser usado para desenvolver diversas coisas, contudo destaca:

- o desenvolvimento de APIs;
- criação de *back-end* de jogos, IoT (Internet das coisas) e apps de mensagens;
- criação de aplicações em tempo real, através do uso de uma web socket, como o Socket.io.

### 2.4.3 Express.js

Almeida (2015, p. 10) afirma que: “O Express estende as capacidades do servidor padrão do Node.js adicionando middlewares e outras capacidades como views e rotas.”.

De acordo com Andrade (2021) o Express.js é um framework, ou seja, um facilitador para o desenvolvimento de diversas aplicações, neste caso para o desenvolvimento *back-end*. O Express.js possui características muito úteis para o desenvolvimento, como um sistema de rotas completo, tratamento de exceções (erros), consegue gerenciar diferentes requisições HTTP e também é muito simples de utilizar, possibilitando a criação de uma API sem precisar de muita complexidade.

### 2.4.4 API

API é um conjunto de definições e protocolos que são usados no desenvolvimento e na integração de aplicações (RED HAT, 2020).

Segundo o mesmo autor, as APIs são como um mediador entre o usuário e o recurso ou serviço web que ele deseja. O usuário se comunica com o sistema através da API, que pode

exigir alguma informação para retornar os dados (como por exemplo um CEP para retornar a previsão meteorológica) ou não exigir nenhum tipo de informação adicional. Não só isso, mas as APIs também oferecem ao desenvolvedor da API (ou organização, empresa, etc.) que mantenham a segurança, controle e a obrigatoriedade de autenticação, permitindo a configuração sobre quem tem acesso e o que pode ser acessado (RED HAT, 2020).

#### 2.4.4 Rest

Saudate (2014, p. 4) afirma que:

REST significa Representational State Transfer (ou Transferência de Estado Representativo, em tradução livre), e é um estilo de desenvolvimento de web services que teve origem na tese de doutorado de Roy Fielding. Este, por sua vez, é co-autor de um dos protocolos mais utilizados no mundo, o HTTP (Hypertext Transfer Protocol).”.

O REST não é um conjunto de restrições de arquitetura, e ele que pode ser implementado pelos desenvolvedores de APIs de maneiras variadas (RED HAT, 2020).

Red Hat (2020, n.p.) afirma que “API REST, também chamada de API RESTful, é uma interface de programação de aplicações (API ou API web) que está em conformidade com as restrições do estilo de arquitetura REST, permitindo a interação com serviços web RESTful.”.

#### 2.4.5 Banco de dados

Banco de dados é uma coleção organizada de informações ou dados estruturadas, geralmente sendo armazenadas eletronicamente em um sistema de computador (ORACLE, 2022).

Um banco de dados geralmente é controlado por um sistema de gerenciamento de banco de dados. A junção dos dados, do gerenciador e dos aplicativos associados a eles é chamada de sistema de banco de dados ou somente banco de dados, como é mais chamado popularmente. A maioria dos bancos de dados utiliza a consulta estruturada (SQL) para escrever e consultar os dados, este que são modelados em linhas e colunas em diversas tabelas diferentes. Com isso, os dados podem ser acessados, gerenciados, modificados, atualizados e organizados (ORACLE, 2022).

Existem diversos tipos de banco de dados (ORACLE, 2022). Os tipos serão representados no Quadro 3.

**Quadro 3 - Tipos de banco de dados**

<b>Bancos de dados relacionais</b>	Dominantes desde a década de 1980, os itens de um banco de dados relacional são organizados em um conjunto de tabelas com colunas e linhas. Essa tecnologia permite com que o acesso as informações estruturadas sejam eficientes e flexíveis.
<b>Banco de dados orientados a objetos</b>	Neste tipo, as informações são representadas na forma de objetos, tal como na programação orientada a objetos.
<b>Banco de dados distribuídos</b>	Consiste em dois ou mais arquivos localizados em sites diferentes. O banco de dados pode ser armazenado em vários computadores, que podem estar localizados em locais físicos distintos ou iguais.
<b>Data warehouses</b>	É um repositório central de dados. Este tipo de banco de dados foi projetado especificamente para consultas e análises rápidas.
<b>Banco de Dados NoSQL</b>	Também conhecido como banco de dados não relacional, este permite que os dados não estruturados e semiestruturados sejam armazenados e manipulados, diferente do relacional que define como todos os dados inseridos devem ser compostos. Este tipo de banco de dados ficou popular conforme os aplicativos web ficaram mais comuns e mais complexos.
<b>Banco de dados gráficos</b>	Armazena dados em termos de entidades e os relacionamentos entre entidades.
<b>Banco de dados OLTP</b>	É um banco de dados rápido e analítico projetado para um grande número de transações feitas por diversos usuários.

Fonte: Elaborado a partir de Oracle (2022).

#### 2.4.5.1 Structured Query Language

O *Structured Query Language*, ou popularmente conhecida apenas como SQL, é uma linguagem de programação usada pela maioria dos bancos de dados relacionais para gerenciar seus dados e controles de acesso (ORACLE, 2022).

Desenvolvido pela primeira vez na IBM nos anos 1970, tendo a Oracle como principal contribuinte, e na mesma época foi criado a implementação do padrão SQL ANSI. Como resultado disso o SQL estimulou muitas extensões de empresas como IBM, Oracle e Microsoft (ORACLE, 2022).

#### 2.4.5.2 PostgreSQL

Soares (2013, p. 434) afirma que:

O PostgreSQL é um banco de dados relacional que vem evoluindo de forma consistente nos últimos anos, agregando funcionalidades antes disponíveis somente em banco de dados pagos, como Oracle e MS SQL Server. Esta é uma das características que mais atraem desenvolvedores ao PostgreSQL, pois além de poderoso, versátil, transaccionalmente seguro (controle total de transações), dispor de stored procedures e triggers, ele é gratuito.

Nas palavras de Souza (2020), o PostgreSQL se destaca pois consegue se adequar a diferentes padrões de conformidade, o que facilita na construção de um banco de dados bem otimizado. O software também oferece segurança no armazenamento de dados e possibilita a restauração de dados caso seja necessário. Além disso, o PostgreSQL consegue se manter estável mesmo com um alto volume de solicitações e com cargas de trabalhos grandes.

## 2.5 FRONT-END DA APLICAÇÃO

Freitas (2021, p.47) afirma que “O lado do cliente é responsável pelas operações na relação cliente-servidor. Este lado realiza a interação com o usuário (*client-side*), também chamado front-end. O cliente pode ser uma aplicação, um navegador Web, uma estação de trabalho, etc.”.

Como aponta Souto (2019), basicamente o *front-end* é a parte visual de um site, e tudo aquilo que o usuário consegue interagir.

### 2.5.1 HTML

Na palavra de Alves (2021), para desenvolver uma página de site é necessária uma linguagem de criação para isso, assim como para o desenvolvimento de outros tipos aplicativos ou programas, assim como o *back-end*. Essa linguagem é chamada de HTML (Hypertext Markup Language), e ela funciona a partir de marcadores, conhecidos como tags.

O mesmo autor aponta que as tags são utilizadas para formatar um tipo de objeto que faz parte da estrutura de um documento HTML, como por exemplo um texto ou uma imagem. Os marcadores são envolvidos pelos sinais < e /> (sendo como abertura e fechamento da tag), apesar de haverem algumas exceções onde não é necessário fechar com /> (ALVES, 2021). A Figura 1 exemplifica a sintaxe geral de um elemento HTML.

**Figura 1 - Sintaxe geral de um elemento HTML**

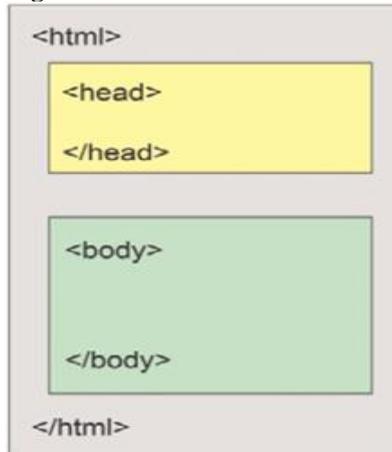


Fonte: Alves (2020, p.7)

De acordo com Alves (2021) os documentos HTML são formados por textos puros, que podem ser criados até por aplicativos de edição de texto – como o próprio bloco de notas do Windows – e, eles são formados por um grupo de seções, onde cada uma é destinada a conter

um tipo de conteúdo específico. Por exemplo, para definir que um o formato HTML, ele deve iniciar com a tag <html> e finalizar com </html>. Outros exemplos que podem ser citados são a tag <head> e </head>, que indica o cabeçalho do documento, ou a tag </body> e </body> que indica o corpo do documento, ou seja, o conteúdo exibido na página. A Figura 2 exemplifica uma estrutura básica de um documento HTML.

**Figura 2 - Estrutura básica de um elemento HTML**



Fonte: Alves (2020, p.7)

### 2.5.2 CSS

O CSS (*Cascading Style Sheets*) serve para estilizar as páginas criadas com códigos feitos nas linguagens HTML, XML ou XHTML. Ele surgiu como uma solução para aprimorar a estética dos sites, visto que o HTML, apesar de conseguir estilizar suas tags sem a necessidade do CSS, não era capaz de suprir todas as necessidades do desenvolvedor na hora de criar padrões de formatação nas páginas, e acabava gerando muita repetição de código (OKUBO, 2021).

Alves (2021) declara que o CSS também possibilitou a reutilização de estilos em vários arquivos HTML diferentes, fazendo com que caso esse estilo seja alterado todas as páginas que o utilizam sejam atualizadas. Os arquivos CSS são compostos por regras de formatação que são associadas a diversos elementos que formam uma página HTML, onde é possível associar um ou mais elementos na mesma regra, sendo necessário apenas separa-los por vírgula. A Figura 3 exemplifica a sintaxe de uma regra CSS.

**Figura 3 - Sintaxe de uma regra CSS**



Fonte: Alves (2021, p.28)

### 2.5.3 React.js

O React é uma biblioteca utilizada para a criação de interfaces de usuário. É comumente utilizado com outras bibliotecas para renderização em certos ambientes. Além disso não é um *framework* e não é exclusivo para a web (MDN CONTRIBUTORS, 2021a).

Como aponta Roveda (2020), o React utiliza a linguagem Javascript, que é uma das mais versáteis e populares linguagens de programação no mundo e tem como objetivo facilitar a interação de diferentes partes de uma página, dessa forma seu funcionamento é através de componentes. Ou seja, desta forma o React divide uma página em vários componentes para formar toda a tela, e assim possibilita que o desenvolvedor trabalhe individualmente em cada um. Esse tipo de funcionamento faz com que o React seja muito flexível tanto na solução quanto na construção de um sistema. Além disso, também acaba por consequência possibilitando muito reaproveitamento de código e padronização de interfaces com os componentes.

### 2.6 NLP

Kovacs (2022, n.p.) afirma que “A NLP — processamento de linguagem natural em português — é o ramo da inteligência artificial que combina linguística computacional, uma modelagem baseada em regras da linguagem humana, e modelos estatísticos de *machine learning* e *deep learning*.”.

De acordo com Silva (2021) através do processamento de linguagem natural é possível que computadores entendam, interpretem e manipulem a linguagem humana, buscando capacitar os computadores para que eles possam lidar com textos escritos por pessoas através da análise do contexto, diferenças de linguagem e até compor textos em resposta. Ou seja, a ideia é que essa tecnologia se assemelhe a um tradutor, garantindo que quando um humano se comunicar com um computador, ele consiga entender claramente o que a pessoa quis dizer.

Para Kovacs (2022) o processamento de linguagem natural pode ser aplicado em diversas áreas, como ilustrado no Quadro 4.

**Quadro 4 - Aplicabilidade da NLP**

<b>Aplicações de tradução</b>	Um dos maiores exemplos, se não o maior, é o Google Tradutor, que utiliza uma NLP muito ampla para não somente trocar uma palavra por outra conforme o idioma, mas também busca entender com maior precisão o significado e o tom do idioma de entrada para então sim traduzi-lo para o idioma de saída.
<b>Assistentes virtuais</b>	Diversas marcas famosas vendem assistentes virtuais atualmente, como a Apple, Amazon e a Google. Estes acessórios usam o reconhecimento da fala para identificar padrões em comandos de voz e geração de linguagem natural para responderem com a ação apropriada ou com algum tipo de comentário.
<b>Chatbots</b>	Os chatbots dão respostas ao usuário conforme recebem entradas de texto digitadas. Alguns chatbots também podem aprender a identificar melhor as pistas ou sinônimos que remetem a uma solicitação humana, e dessa forma facilitar seu entendimento com o decorrer do tempo e assim fornecer melhores respostas.

Fonte: Adaptado de Kovacs (2022).

### 2.6.1 Node NLP

O NLP.js ou Node NLP, é um utilitário geral de linguagem natural feito para o Node.js, onde com ele é possível fazer o processamento de mensagens, sendo possível realizar algumas coisas como identificar o idioma de uma frase, fazer uma análise de sentimentos e até mesmo gerenciar diversos idiomas ao mesmo tempo, onde entre eles a biblioteca suporta 40 de forma nativa e mais 104 a partir da integração com o BERT (AXA GROUP, 2022).

Para iniciar um projeto com o Node NLP, após as configurações iniciais, é necessário que criar uma série de mensagens chave para uma intenção e quais respostas o bot deverá dar ao identificá-la. Isso pode ser feito tanto por um arquivo externo JSON quanto adicionando tudo no código-fonte, e após configurado o bot poderá treinar as novas intenções e estará pronto para conversar (AXA GROUP, 2022).

Além disso, a biblioteca também permite com que seja possível criar um projeto para processar mensagens rapidamente, oferecendo algumas alternativas nativas para testar ou até mesmo usar o bot: pelo terminal, por um chat web que a própria biblioteca cria ou usando o chat do framework da Microsoft, que roda integrado com o Node.js (AXA GROUP, 2022).

### 2.7 LGPD

A Lei Geral da Proteção de Dados, Lei nº 13.709/2018 de 14 de agosto de 2018, decretada com o intuito de proteger os direitos de privacidade bem como de liberdade da personalidade do indivíduo. Refere-se ao tratamento dos dados pessoais, sejam eles físicos ou digitais, através de uma pessoa física ou jurídica pertencente ao direito público ou privado (BRASIL, 2021a).

BRASIL (2021<sup>a</sup>, n.p.) afirma que:

Tema fundamental trabalhado pela Lei, o tratamento de dados diz respeito a qualquer atividade que utiliza um dado pessoal na execução da sua operação, como, por exemplo: coleta, produção, recepção, classificação, utilização, acesso, reprodução, transmissão, distribuição, processamento, arquivamento, armazenamento, eliminação, avaliação ou controle da informação, modificação, comunicação, transferência, difusão ou extração.

De acordo com Brasil (2021b), a lei estabelece os direitos dos titulares dos dados pessoais, os quais devem ser garantidos e protegidos durante todo seu tratamento pela entidade/órgão. A Lei da Geral da Proteção de Dados é composta pela estruturação de alguns princípios, sendo eles apresentados no Quadro 5.

**Quadro 5 - Princípios da LGDP**

<b>Finalidade:</b>	A realização do tratamento deve ocorrer para propósitos legítimos, específicos, explícitos e informados ao(à) titular, sem possibilidade de tratamento posterior de forma incompatível com essas finalidades;
<b>Adequação:</b>	A compatibilidade do tratamento deve ocorrer conforme as finalidades informadas ao(à) titular, de acordo com o contexto do tratamento;
<b>Necessidade:</b>	O tratamento deve se limitar à realização de suas finalidades, com abrangência dos dados pertinentes, proporcionais e não excessivos em relação às finalidades do tratamento de dados;
<b>Livre Acesso:</b>	É a garantia dada aos(às) titulares de consulta livre, de forma facilitada e gratuita, à forma e à duração do tratamento, bem como à integralidade de seus dados pessoais;
<b>Qualidade dos Dados:</b>	É a garantia dada aos(às) titulares de exatidão, clareza, relevância e atualização dos dados, de acordo com a necessidade e para o cumprimento da finalidade de seu tratamento;
<b>Transparência:</b>	É a garantia dada aos(às) titulares de que terão informações claras, precisas e facilmente acessíveis sobre a realização do tratamento e os respectivos agentes de tratamento, observados os segredos comercial e industrial;
<b>Segurança:</b>	Trata-se da utilização de medidas técnicas e administrativas qualificadas para proteger os dados pessoais de acessos não autorizados e de situações acidentais ou ilícitas de destruição, perda, alteração, comunicação ou difusão;
<b>Prevenção:</b>	Compreende a adoção de medidas para prevenir a ocorrência de danos por causa do tratamento de dados pessoais;
<b>Não discriminação:</b>	Sustenta que o tratamento dos dados não pode ser realizado para fins discriminatórios, ilícitos ou abusivos;
<b>Responsabilização e prestação de contas:</b>	Demonstração, pelo Controlador ou pelo Operador, de todas as medidas eficazes e capazes de comprovar o cumprimento da lei e a eficácia das medidas aplicadas.

Fonte: Adaptado de Brasil (2021b).

### 2.7.1 Dados anonimizados

De acordo com Serpro (2022), os dados anonimizados são aqueles que inicialmente eram de uma pessoa, mas que passou por algumas etapas de tratativas para desvincular os

mesmos a essa pessoa, por conseguinte, por ser um dado anonimizado, a LGPD não se aplicará sobre ele.

Segundo o mesmo autor, um dado só poderá ser intitulado como anonimizado apenas se não permitir que em algum momento, por parte técnica ou de alguma outra forma, possa ser identificado a pessoa portadora desse dado. Caso esse dado for identificado em algum momento, o mesmo não será originalmente um dado anonimizado, sendo apenas cognominado como “pseudonimizado”, e dessa forma, estará sujeito a LGPD (SERPRO, 2022).

Serpro (2022) afirma que os dados anonimizados são fundamentais para a área da tecnologia, principalmente na inteligência artificial, na formação de cidades inteligentes, entre outros. Também aconselham que as organizações, sendo elas públicas ou privadas, realizem a anonimização dos dados pessoais, dessa forma garantindo uma segurança da informação na organização, e também gerando mais confiabilidade em seus serviços.

### 3. METODOLOGIA DA PESQUISA

Este trabalho de conclusão de curso se caracteriza como uma pesquisa aplicada e descritiva, pois foi desenvolvido um protótipo de um *chatbot* para encontrar profissionais da saúde, tendo também como objetivo verificar se o protótipo atende as necessidades básicas para o usuário.

O trabalho buscou responder o seguinte problema: Como ajudar as pessoas a encontrarem profissionais da saúde especializados em determinadas áreas, tanto pelo celular quanto pelo computador, sendo que geralmente eles não ficam indexados no Google? Para a construção dessa pesquisa foi realizada a revisão da literatura onde foram apresentados os conceitos de todas as tecnologias necessárias para o desenvolvimento da aplicação. No capítulo de desenvolvimento foi realizada uma análise com o intuito de entender as necessidades que a aplicação deve suprir para em seguida dar início aos processos de modelagem e implementação. A modelagem e a estrutura do banco de dados foram feitas para o PostgreSQL. Já na etapa de implementação, o desenvolvimento *back-end* foi desenvolvido com a tecnologia Node.js e o *front-end* foi desenvolvido com o React.js de forma responsiva, ou seja, a aplicação pode ser usada em diversos dispositivos, como por exemplo computadores, notebooks, celulares ou tablets. Além disso também foi utilizado a tecnologia Typescript para auxiliar no desenvolvimento tanto do *back-end* quanto do *front-end*.

#### 3.1 ESTADO DA ARTE

Nessa seção será apresentada uma das ferramentas disponíveis no mercado, que oferece um serviço similar ao proposto no protótipo. Na análise a seguir, apontaremos algumas vantagens e desvantagens da ferramenta Iubem.

##### 3.1.1 Iubem

O Iubem é uma ferramenta gratuita que busca fazer uma ponte entre os pacientes e os profissionais da saúde. Para isso a empresa oferece dois aplicativos, um sendo voltado para o paciente e outro para o profissional (IUBEM, 2022).

Como paciente (Figura 4), o usuário pode procurar profissionais de saúde conforme suas necessidades e na localização que desejar, de forma rápida e prática. Além disso, a ferramenta também oferece a possibilidade de pagamento da consulta direto pelo aplicativo (após a consulta ser efetivada) e o cadastro da sua ficha médica (IUBEM, 2022).

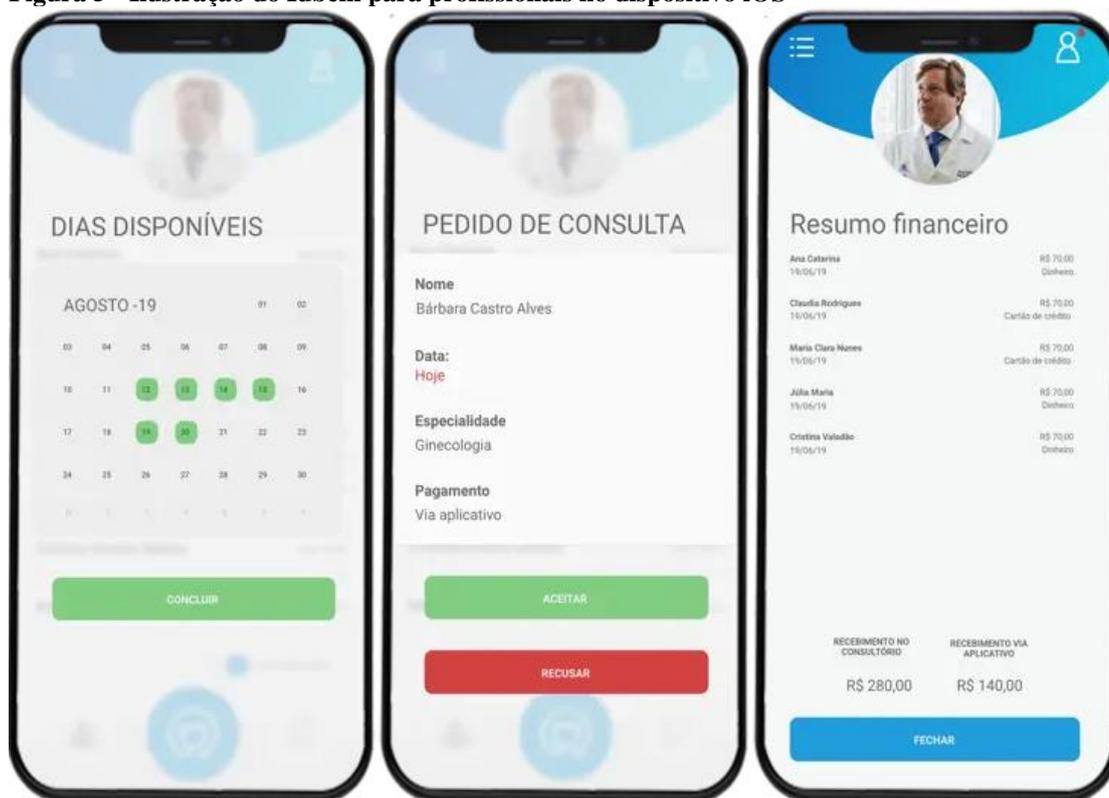
**Figura 4 - Ilustração do Iubem para pacientes no dispositivo iOS**



Fonte: App Store (2022).

Já o profissional (Figura 5) pode cadastrar sua agenda, disponibilizar horários de atendimentos particulares ou por convênio, e através do algoritmo baseado nos sistemas de mobilidade a ferramenta faz o envio das solicitações aos profissionais de saúde conforme os pré-requisitos selecionados pelo paciente. A única cobrança feita pela ferramenta é na efetivação da consulta, aonde 15% é descontado do valor da consulta. Caso o paciente realize o pagamento no consultório, esse valor é cobrado via boleto (IUBEM, 2022).

**Figura 5 - Ilustração do Iubem para profissionais no dispositivo iOS**



Fonte: App Store (2022).

A aplicação possui versões para Android e iOS, abrangendo grande parte dos dispositivos disponíveis no mercado, possibilitando o uso de qualquer lugar tanto para o paciente quanto para o profissional.

As duas versões oferecem uma interface agradável ao usuário, e tem uma usabilidade simples e intuitiva. Para o paciente em poucas etapas é possível se cadastrar e realizar a busca de um profissional desejado, sem mistério. Já o profissional, após realizar seu cadastro rapidamente consegue vincular sua conta bancária, organizar sua disponibilidade e verificar sua agenda de pacientes, além de receber novos pedidos de consultas.

Contudo, uma das desvantagens é que por ser um aplicativo restrito a dispositivos móveis, tem um público alvo menor. Outro ponto é que como o aplicativo não tem uma versão web e isso pode diminuir a agilidade do profissional no dia a dia de trabalho.

## **4. CHATBOT ONLINE PARA ENCONTRAR PROFISSIONAIS DA SAÚDE**

Neste capítulo serão abordados assuntos referentes à construção do protótipo, demonstrando a análise do projeto e a implementação do trabalho.

### **4.1. ANÁLISE**

Conforme citado na revisão da literatura, a engenharia de software é muito importante desde a criação do projeto, sendo utilizada para entender a necessidade de negócio do cliente e moldar as formas para alcançar este objetivo. Através da análise foi possível mapear tudo que deverá ser desenvolvido e desta forma facilitar a manutenção do software, pois é mais fácil para identificar e corrigir os problemas.

Além disso, comumente se utiliza da engenharia de software e suas ferramentas, para realizar a validação da ideia, uma vez que ao descrever e diagramar diversas modificações são aplicadas ao projeto, de forma que o mesmo fique mais aderente ao negócio e a tecnologia.

#### **4.1.1. Visão Geral**

O presente trabalho tem como principal meta ajudar as pessoas a encontrarem profissionais da saúde através da utilização do chat, passando algumas informações para o robô para que dessa forma o sistema possa retornar ao usuário as informações encontradas, sendo elas o nome do médico ou clínica, especialidades, contatos e a localização.

Utilizando a aplicação os usuários terão mais facilidade e agilidade para encontrar profissionais, sem ter que ligar para alguma clínica pedindo informação ou buscando em listas telefônicas.

Para o profissional da saúde, ele poderá se cadastrar no sistema e informar todas as suas especialidades, bem como poderá atender na rede pública ou privada e indicar seus principais convênios. Dessa forma já ficará disponível na base de dados do robô de forma rápida e ágil, com isso além de ajudar as pessoas o profissional também pode obter mais clientes.

#### **4.1.2 Comparação do Protótipo com o Estado da Arte**

Foram realizadas algumas comparações entre o protótipo deste trabalho e a aplicação listada na seção de Estado da Arte, apresentando alguns recursos existentes no sistema avaliado. O Quadro 6 representa a diferença dos recursos de cada aplicação.

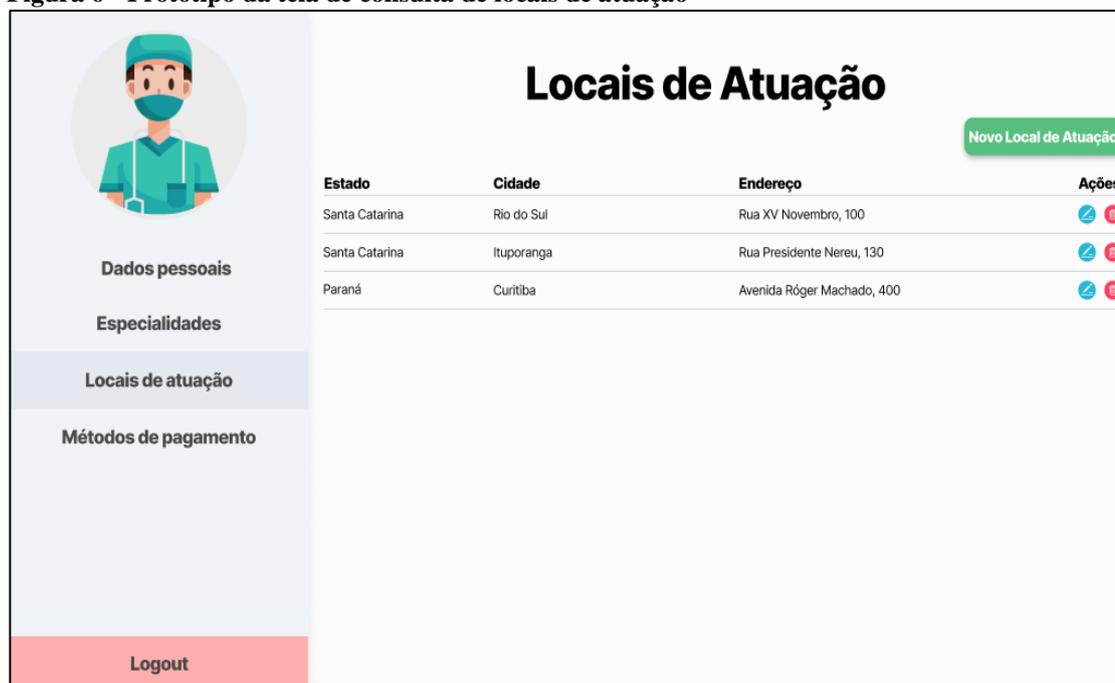
**Quadro 6 - Recursos no sistema avaliado**

Funcionalidade	Chatbot Online	Iubem
Busca através de um sistema conversacional	X	
Controle sobre a agenda do profissional		X
Aplicativo específico para dispositivos móveis		X
Sistema possibilita o agendamento do paciente com o médico		X
Sistema permite o cadastro de ficha médica		X
Sistema permite a busca de profissionais por região e especialidade	X	X
Sistema possibilita o pagamento da consulta através do aplicativo		X
Busca por profissionais totalmente de forma anônima	X	

Fonte: Acervo do Autor.

### 4.1.3. Prototipação

Para realizar o desenvolvimento das telas da aplicação, foi criado os protótipos de todas as telas através da ferramenta Figma. As Figuras 6 e 7 representam os protótipos das telas de consulta dos locais de atuação e manutenção das especialidades respectivamente.

**Figura 6 - Protótipo da tela de consulta de locais de atuação**

Fonte: Acervo do Autor.

**Figura 7 - Protótipo da tela de manutenção das especialidades**

Fonte: Acervo do Autor.

A Figura 8 representa o protótipo da tela inicial do sistema com o *chatbot* aberto onde o robô iniciou uma nova conversa.

**Figura 8 - Tela inicial com o chatbot aberto e o robô iniciando a conversa**

Fonte: Acervo do Autor.

## 4.2. REGRAS DE NEGÓCIO

As regras de negócio são responsáveis por indicar quais serão as normas do negócio que deverão ser implementadas, para que dessa forma a aplicação funcione da maneira correta. No Quadro 7 serão listadas as principais regras de negócio do protótipo.

**Quadro 7 - Regras de Negócio**

<b>Número</b>	<b>Descrição</b>
RN01	O profissional deve cadastrar seus dados pessoais, suas especialidades e seus locais de atuação para participar da plataforma. O profissional pode informar mais de uma especialidade em seu cadastro, contudo não pode conter informações duplicadas.
RN02	Para que o cadastro do profissional seja efetuado, é necessário que o mesmo informe um número de registro de classe.
RN03	O profissional pode ter vários convênios médicos em cada local de atuação cadastrado. Por exemplo, um profissional pode atender tanto na UNIMED quanto no SUS.
RN04	O profissional pode ter vários métodos de pagamento vinculados em cada local de atuação.
RN05	Devido a Lei Geral de Proteção de Dados (LGPD), para que o profissional efetue seu cadastro e faça parte da plataforma, ele precisará aceitar todos os termos e condições do site, pois seus dados profissionais serão públicos e exibidos pelo robô no <i>chatbot</i> .
RN06	O profissional pode alterar seus dados profissionais ou remove-los. Isso acontece pois o mesmo pode não realizar mais atendimentos em uma especialidade específica ou ter alterações em algum vínculo empregatício, como o local de atendimento, convênio ou na forma de pagamento e o usuário deve conseguir visualizar seus dados sempre mais recentes.
RN07	Após cadastrado e validado, o profissional não poderá editar seu CPF, assim como não poderá criar um novo cadastro usando o mesmo dado.
RN08	O sistema não poderá armazenar buscar ou resultados de buscas quando o usuário estiver identificado, ou se armazenar, deverá fazê-lo de forma anonimizada, conforme preconiza a LGPD.
RN09	O sistema deverá apresentar-se ao usuário de forma que permita a fácil compreensão do uso, opções disponíveis, forma de acesso, interatividade e por fim o fácil aprendizado, uma vez que não haverá material de treinamento disponível.
RN10	O login do profissional é de uso exclusivo e intransferível dele. Nenhum outro usuário deverá conseguir acessar os dados do profissional com o seu login. O login deverá conter mecanismos de segurança suficientes para que o sistema impeça o uso ilegal.
RN11	O nome de usuário deve ser único no sistema.
RN12	O profissional poderá informar diversos locais de atuação, cada qual com seu próprio endereço e especialidades. Além disso, o profissional poderá editar ou remover os locais de atuação a qualquer momento.
RN13	Para encontrar profissionais será necessário sempre definir uma região de abrangência.
RN14	O profissional pode editar seus dados pessoais a qualquer momento.
RN15	O <i>chatbot</i> deve ser acessível a todos os usuários (incluindo usuários anônimos), e sempre que um usuário iniciar uma nova conversa o robô deve mandar a mensagem de boas-vindas e solicitar os dados iniciais para realizar a busca de um profissional.
RN16	O usuário pode fazer login e logout da plataforma a hora que quiser.
RN17	O profissional pode excluir uma especialidade sempre que quiser.

Fonte: Acervo do Autor.

## 4.3 REQUISITOS

Os requisitos são recursos do sistema que podem ou devem estar presentes no protótipo para que ele cumpra com o seu propósito. O Quadro 8 representa os requisitos não funcionais, ou seja, os requisitos que estão relacionados a características de qualidade e ao uso da aplicação.

**Quadro 8 - Requisitos não funcionais**

Número	Descrição
RNF01	O sistema deverá salvar a senha do usuário de forma criptografada no banco de dados para que haja maior segurança na utilização do sistema.
RNF02	O sistema deverá ser construído com uma paleta de cores pastéis. Dessa forma, a utilização do sistema fica mais agradável aos usuários sem as cores chamativas.
RNF03	Consistência visual (as telas devem ter a mesma aparência, os botões e a interação devem ser nos mesmos locais). O objetivo é facilitar o aprendizado do usuário.

Fonte: Acervo do Autor.

No Quadro 9 serão listados os requisitos funcionais do protótipo, estes que são referentes às funcionalidades necessárias para a realização das atividades pelo sistema, ou seja, as funções do sistema que estarão disponíveis para o usuário. A relação com as regras de negócio tem como objetivo detalhar a rastreabilidade entre cada função relativamente ao negócio.

**Quadro 9 - Requisitos funcionais**

Número	Nome	Descrição	Regras de Negócio relacionadas
RF01	Cadastro dos dados pessoais do profissional	O sistema deverá solicitar ao profissional, que desejar participar da plataforma, o cadastro através do site, informando seus dados pessoais.	RN01
RF02	Tela inicial	O sistema deverá conter uma tela inicial com uma breve descrição da aplicação, para poder auxiliar o usuário em como utilizar o <i>chatbot</i> . Nesta mesma tela, deverá conter um botão para iniciar a conversa com o robô.	RN09
RF03	Manutenção de dados pessoais	O sistema deverá permitir que o profissional edite suas informações pessoais.	RN14
RF04	Cadastro único do profissional	O sistema deverá impedir que um profissional tente se registrar com um CPF já cadastrado na plataforma.	RN07
RF05	Cartão de visitas do profissional	O sistema deverá exibir os dados do profissional encontrado pelo robô, em forma de cartão de visitas. Nesse cartão deverá conter informações como o nome do profissional, a especialidade, convênios, local de atuação e formas de pagamento.	RN06
RF06	Tela de login	O sistema deverá conter um login. Os dados necessários para entrar devem ser o nome de usuário e a senha.	RN10
RF07	Senha do usuário	O sistema deverá dispor de recurso para validar a força da senha para que ela contenha pelo menos 8 dígitos, e siga os padrões já estabelecidos por sistemas consolidados.	RN10
RF08	Cadastro de especialidades	O sistema não deverá aceitar que seja informado números de registros duplicados no cadastro de especialidades.	RN01
RF09	Visualização dos cartões de visita dos profissionais	O sistema deverá mostrar os profissionais em forma de carrossel, sempre que o robô encontrar as informações solicitadas pelo usuário no <i>chatbot</i> . Essas informações deverão ser apresentadas em forma de um cartão de visitas, contendo todas as informações cadastradas pelo profissional na especialidade selecionada.	RN15
RF10	Termos de uso da plataforma	O sistema deverá perguntar ao profissional se ele aceita os termos e condições do site para poder realizar o cadastro do mesmo.	RN05

RF11	Cadastro do número de registro profissional	O sistema deverá permitir que o profissional informe qual é a sua entidade de classe bem como o número, com isso, também deverá validar o número de identificação de acordo com a entidade de classe do profissional (exemplos: CRM, CRF, etc.).	RN02
RF12	Cadastro do usuário	O sistema deverá solicitar apenas o nome de usuário e uma senha, para que posteriormente seja possível fazer login e finalizar seu cadastro na plataforma.	RN11
RF13	Manutenção de locais de atuação	O sistema deverá possibilitar a consulta de todos os locais de atuação do profissional, tal qual editar ou deletar os registros.	RN12
RF14	Manutenção de especialidades	O sistema deverá possibilitar a consulta e a exclusão de especialidades.	RN01, RN17
RF15	Bloquear campo CPF	O sistema deverá bloquear o campo CPF e impossibilitar que o usuário edite essa informação.	RN07
RF16	Chatbot na tela inicial	O sistema deverá conter um <i>chatbot</i> acessível através da tela inicial da aplicação. O bot deverá ser criado com utilizando a biblioteca Node NLP.	RN15
RF17	Logout da plataforma	O sistema deverá permitir que o usuário faça logout da plataforma através do painel do profissional e pela tela inicial.	RN16
RF18	Cadastro de locais de atuação	O sistema deverá possibilitar que o profissional cadastre vários locais de atuação. Nele o profissional também pode informar em quais convênios ele atende e quais são as formas de pagamento nos locais em que atua.	RN03, RN04

Fonte: Acervo do Autor.

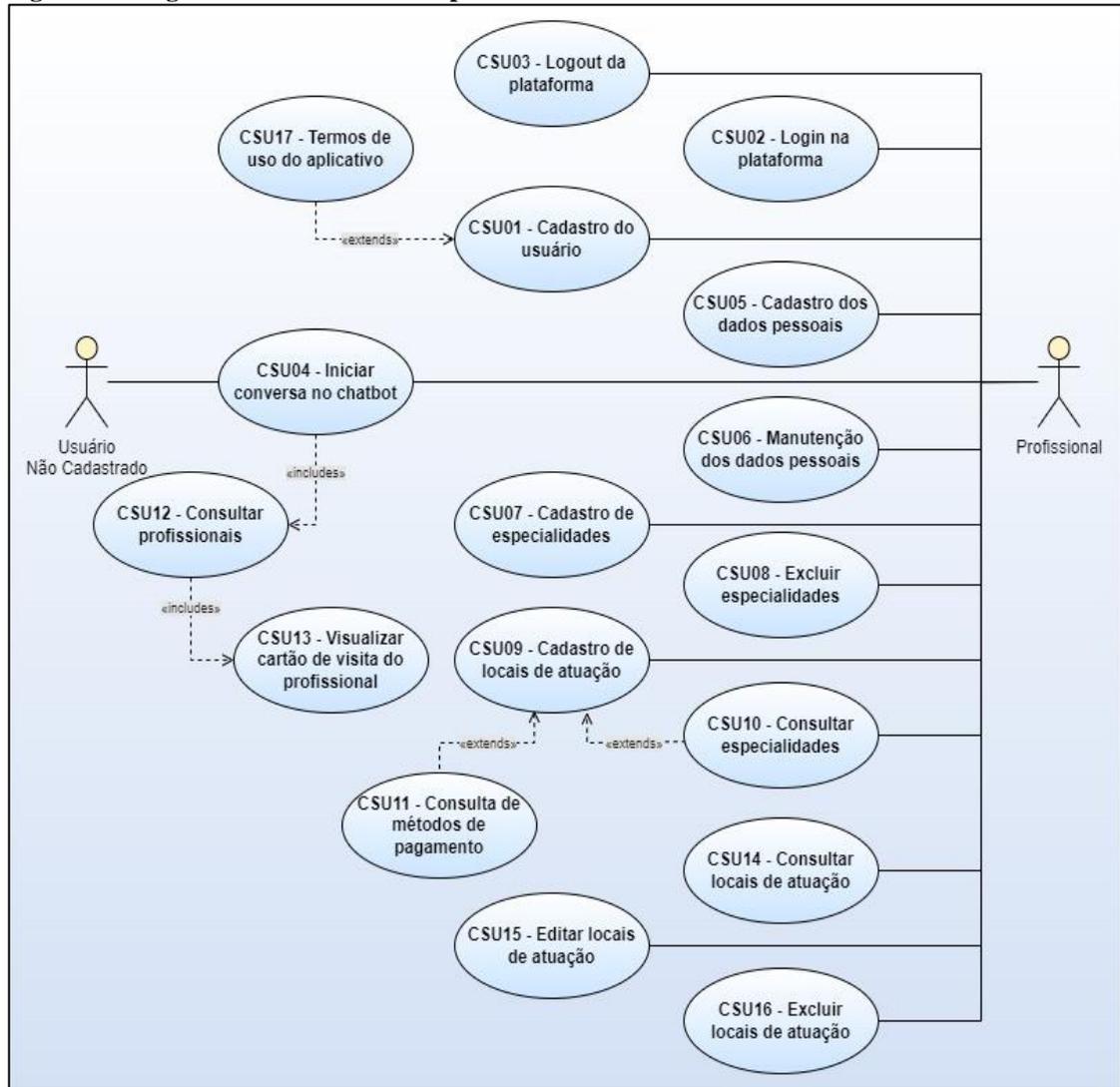
#### 4.4 DIAGRAMAS

Os diagramas a seguir representam os processos de utilização do sistema e casos de uso, ou seja, as ações realizadas pelos atores envolvidos. Com isso, o objetivo é melhorar o entendimento e a visualização dos principais recursos do sistema.

Uma pessoa que desejar participar do processo do sistema, ou seja, um profissional da saúde, poderá se cadastrar na plataforma. Para isso será necessário informar seus dados pessoais e profissionais para que dessa forma o robô possa mostrar seu cartão de visitas para os usuários que acessarem o *chatbot*. Junto a isso, um usuário comum terá acesso apenas a página inicial onde ele poderá iniciar um chat com o robô para buscar profissionais de determinados segmentos da saúde em determinadas regiões.

A Figura 9 representa o diagrama dos casos de uso envolvendo todas as ações disponíveis para o profissional e para o usuário não cadastrado na plataforma.

**Figura 9 - Diagrama de caso de uso do profissional e do usuário não cadastrado.**



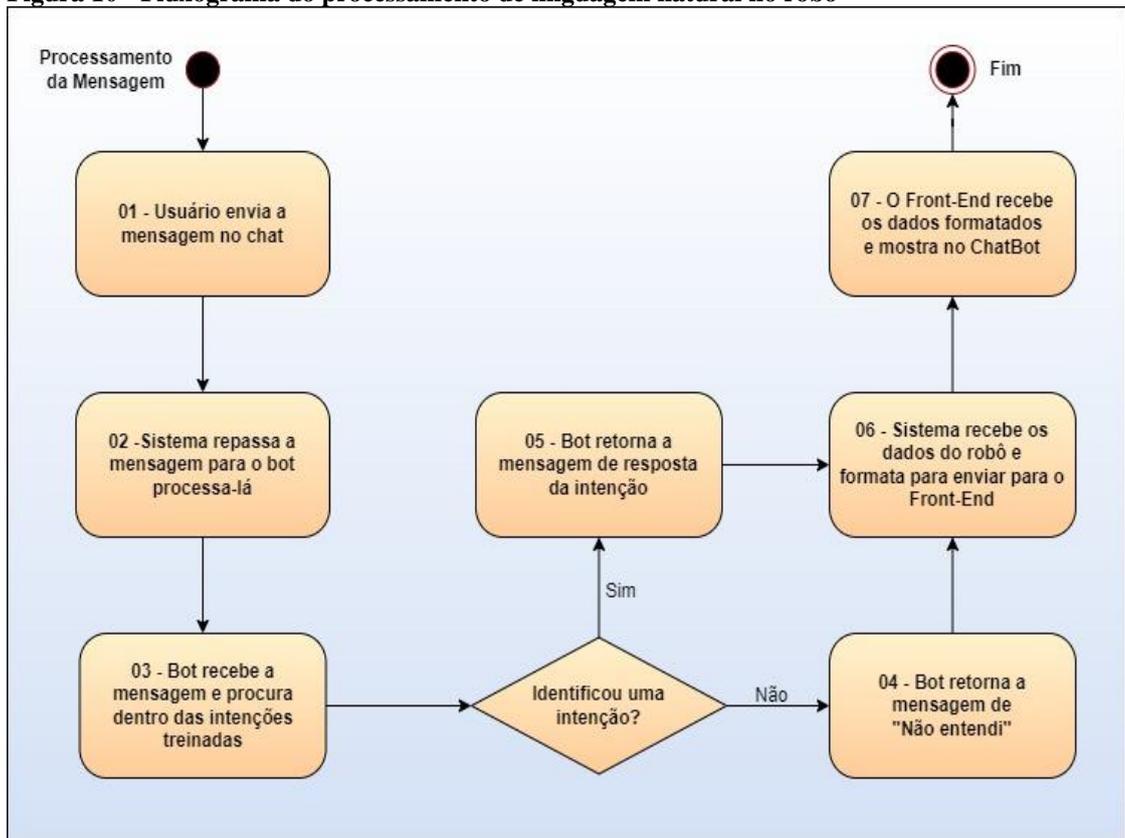
Fonte: Acervo do Autor.

Dentre as funcionalidades do sistema, a principal é o *chatbot* que pode ser acessado por qualquer pessoa. O chat utiliza um bot para conversar com o usuário, podendo conversar com vários usuários simultaneamente. Não foi desenvolvida uma alimentação automática do *PLN* pois isso poderia influenciar negativamente as conversas.

Um exemplo não muito recente aconteceu com o robô do app SimSimi que continha uma inteligência artificial colaborativa, ou seja, os usuários conseguiam alimentar o entendimento do robô, contudo em 2018 ele foi desativado pois os usuários acabaram ensinando respostas maliciosas, fazendo com que o robô tivesse ensinamentos ruins e desrespeitosos (DA REDAÇÃO, 2018).

Com a Figura 10, podemos observar o fluxograma do processamento da linguagem natural do robô.

**Figura 10 - Fluxograma do processamento de linguagem natural no robô**



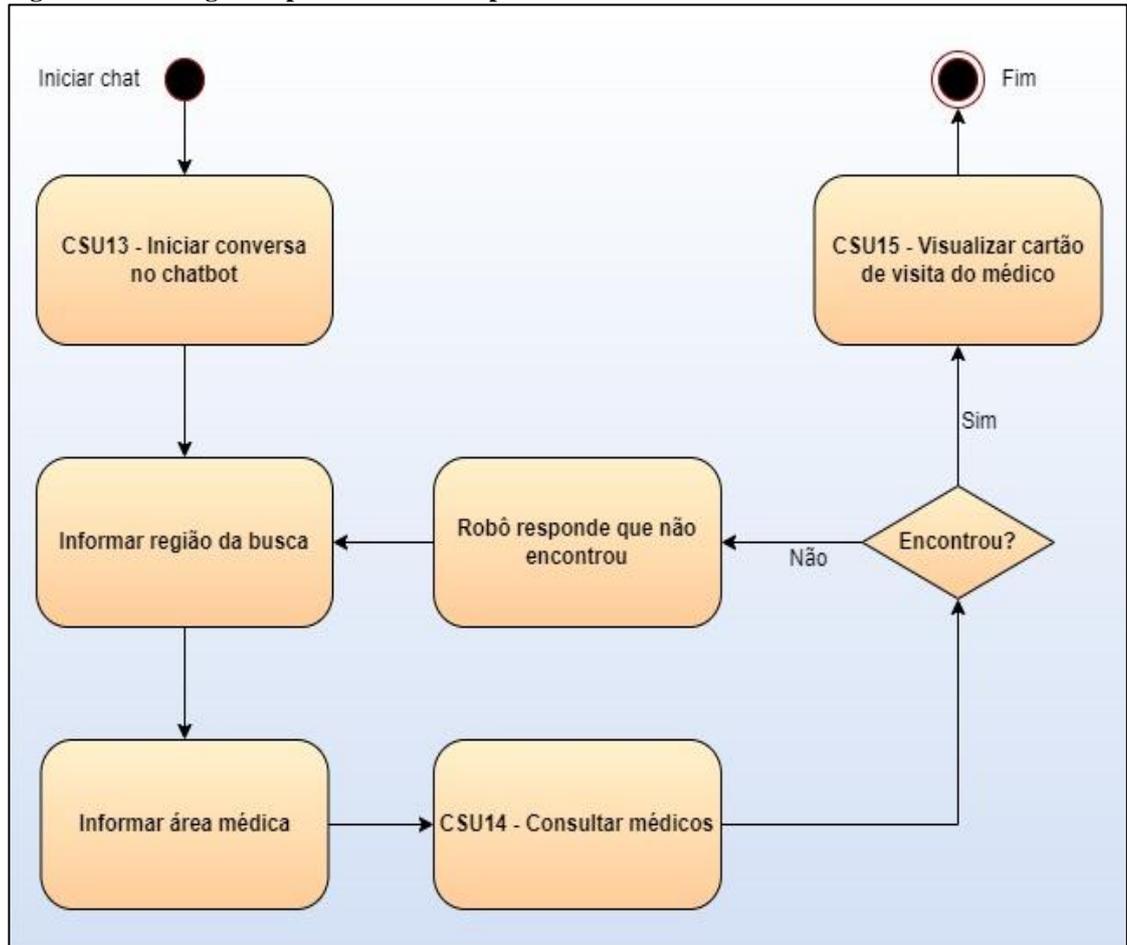
Fonte: Acervo do Autor.

Conforme a Figura 10, atividade 3, percebe-se que é feito o processamento da mensagem recebida do usuário e então identificar qual a intenção do texto. Para fazer isso através do Node NLP, foi necessário definir palavras chaves em arquivos de configurações do robô, que realiza a busca utilizando a mensagem recebida e faz a verificação de qual intenção tem mais probabilidade de ser a desejada pelo usuário, com base nas suas intenções cadastradas e treinadas, e dessa forma enviará a mensagem de resposta. Caso o robô não encontre uma intenção treinada que seja próxima a que o usuário enviou, por padrão o robô retornará que não entendeu.

Apesar do robô ter um processamento de linguagem natural e, com isso conseguir trocar algumas palavras com o usuário, seu principal objetivo é fazer a busca dos profissionais da saúde utilizando as informações fornecidas pelo usuário, por causa disso seus principais conhecimentos são relacionados com a intenção do usuário buscar um profissional, para que caso encontre resultados, exiba-os em forma de cartão de visita.

Na Figura 11, é possível visualizar o fluxograma para solicitar um profissional da saúde para o robô.

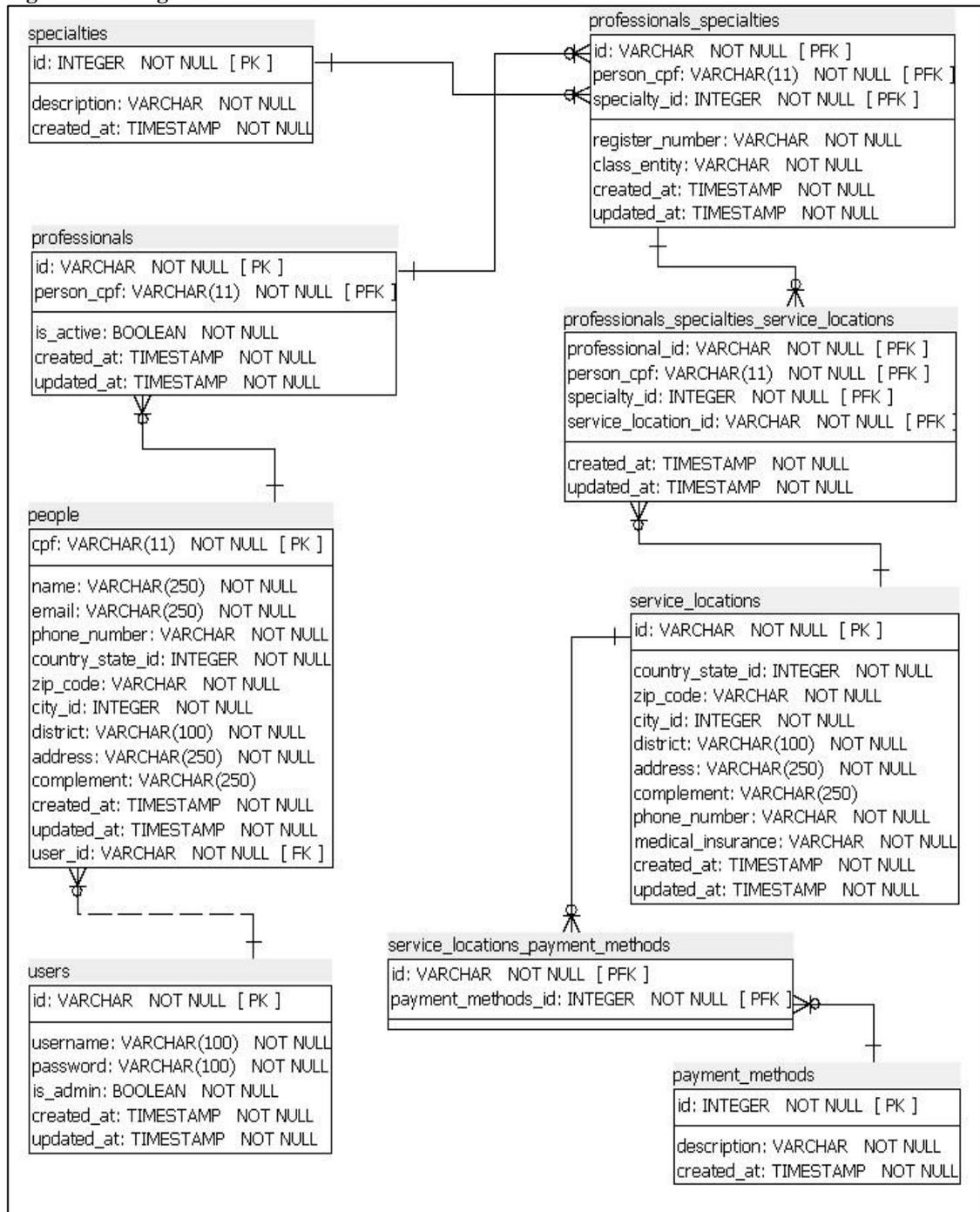
**Figura 11 - Fluxograma para solicitar um profissional da saúde**



Fonte: Acervo do Autor.

A Figura 12 representa o Diagrama de Entidade e Relacionamento (DER), detalhando as tabelas e seus relacionamentos no banco de dados do protótipo.

**Figura 12 - Diagrama de modelo entidade e relacionamento**



Fonte: Acervo do Autor.

## 4.5 IMPLEMENTAÇÃO

Como citado anteriormente, para o desenvolvimento do protótipo foram utilizadas tecnologias voltadas para aplicações *web*, sendo a principal o Javascript, ao qual serve como base para todas as outras tecnologias, atuando no desenvolvimento das telas e páginas da

aplicação *web*, na configuração e ensinamento do robô e, no controle de dados do lado do servidor, ao qual executa a *API*.

Sobre o Javascript foi usado o Typescript, que é um *superset* de tipagens usado apenas no desenvolvimento para melhorar a produtividade e legibilidade, pois permite que o código contenha uma sintaxe mais moderna com checagem de tipos em conjunto com o uso da sintaxe original do Javascript e seus recursos. Essa ferramenta está presente na aplicação *web* e na de controle de dados, porém não foi utilizada no projeto de processamento de linguagem natural.

No desenvolvimento da *API* foi utilizado o Node.js para criar o *back-end* pois, este *framework* permite que a versatilidade do Javascript seja usada no lado servidor. Junto a essa ferramenta também foi adicionado o Express.js que tem como função fornecer um sistema de rotas HTTP, que podem ser gerenciadas facilmente e dessa forma receber requisições enviadas do cliente.

Na Figura 13 será representado o uso do Express na criação de rotas na aplicação.

**Figura 13 - Criação de rotas com Express.js**

```
import { Router } from 'express';
import ListSpecialtiesController from 'useCases/listSpecialties/ListSpecialtiesController';

const specialtiesRoutes = Router();

const listSpecialtiesController = new ListSpecialtiesController();

specialtiesRoutes.get('/', listSpecialtiesController.handle);
```

Fonte: Acervo do Autor.

Em relação ao desenvolvimento *web* da aplicação foi utilizado o React.js, ao qual permite o uso de Javascript para envolver a lógica dos componentes, além do HTML e a estilização CSS. Além disso, também foi utilizado o Axios, este que é uma biblioteca usada para fazer as requisições para o *back-end*, ou seja, enviar ou requisitar informações ao servidor.

## 4.6 UTILIZAÇÃO E FUNCIONAMENTO

Nessa seção será apresentado a estrutura de distribuição da aplicação, exemplificando de forma visual como as tecnologias usadas no projeto estão dispostas. Além disso, também será abordado mais detalhes sobre o funcionamento do sistema.

### 4.6.1 Estrutura de distribuição da aplicação

O protótipo foi construído usando diversas tecnologias sendo as principais são o Node.js e o React.js, ambas sendo tecnologias Javascript. Além disso, foi necessário utilizar bibliotecas

adicionais para complementar o uso das tecnologias, como componentes React ou encriptadores. Entretanto foi indispensável o uso do PostgreSQL, pois é necessário salvar os dados de uma forma fácil e segura. Quanto à manutenção do banco de dados, foi instalado o PgAdmin 4, pois facilita o trabalho durante o desenvolvimento e possibilitar a consulta de dados, adicionar configurações e fazer outros tratamentos de uma forma mais visual.

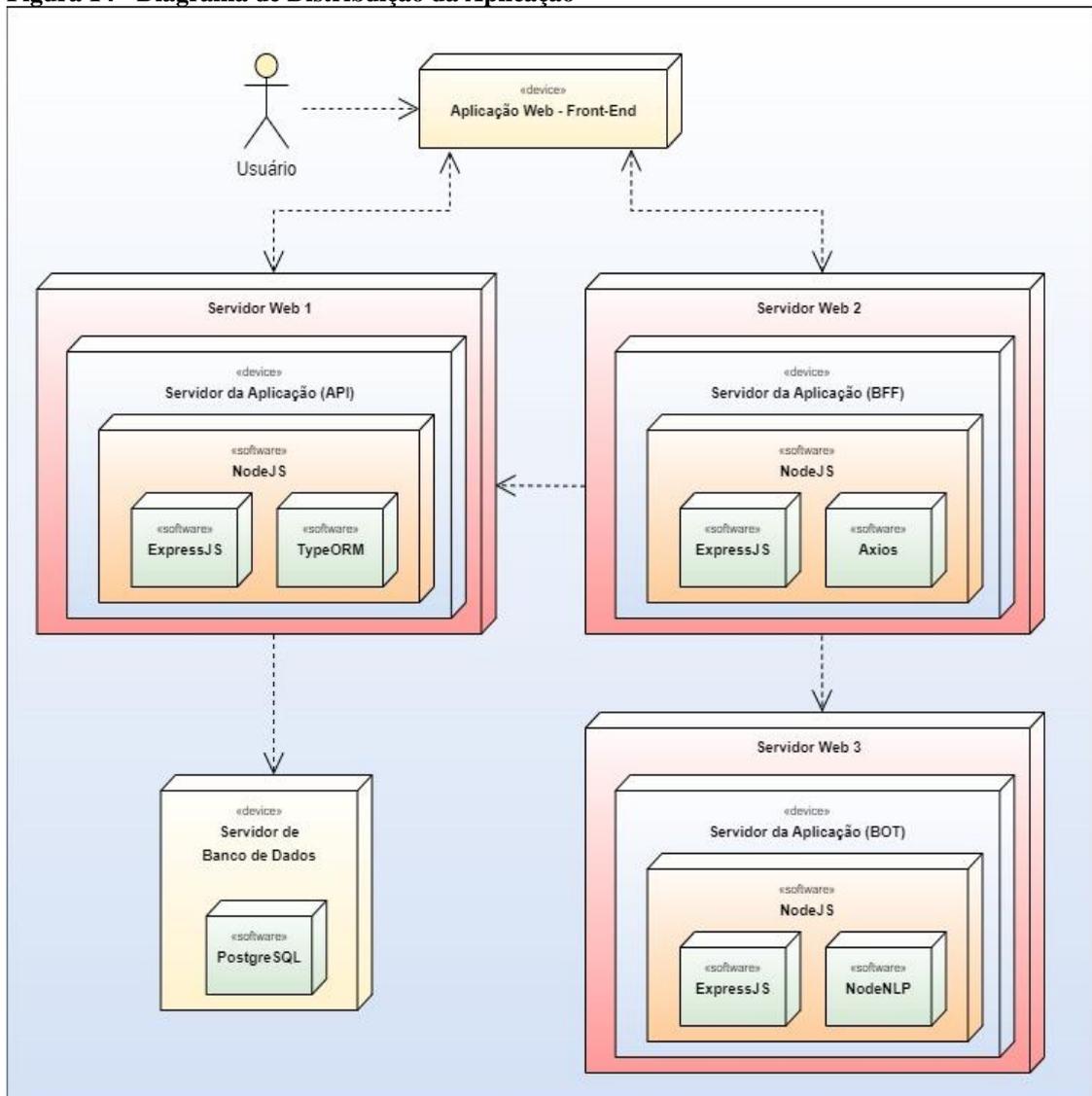
A aplicação *web* foi desenvolvida toda em React. Essa tecnologia é escrita toda em Javascript, e ele basicamente separa a tela em vários componentes, permitindo que seja possível trabalhar sobre cada um de forma individualizada. Essa tratativa possibilita o reaproveitamento de códigos e uma melhor padronização de interfaces, fazendo com que haja ganhos tanto no desenvolvimento e quanto na manutenção do projeto.

Em relação ao servidor da aplicação, a tecnologia principal é o Node.js, que é um *framework* que permite o uso de todas as versatilidades do Javascript no lado do servidor, ou seja, no *back-end*. Todos os servidores utilizam do Express.js para realizar os tratamentos das requisições HTTP que são feitas para a *API*, e após isso os controladores fazem as alterações necessárias no banco de dados.

Além do servidor principal, foram criados outros dois servidores relacionados ao *chatbot*, um *BFF (Back-end For Frontend)* que serve para realizar todos os tratamentos necessários para apresentar os dados no *chatbot*, sem que essa responsabilidade seja do servidor principal, e um servidor para o *bot*, ao qual contém todas as configurações e aprendizados do robô que é utilizado no chat.

Na Figura 14 será representado o diagrama de distribuição, e através dele será possível visualizar as informações citadas anteriormente.

**Figura 14 - Diagrama de Distribuição da Aplicação**



Fonte: Acervo do Autor.

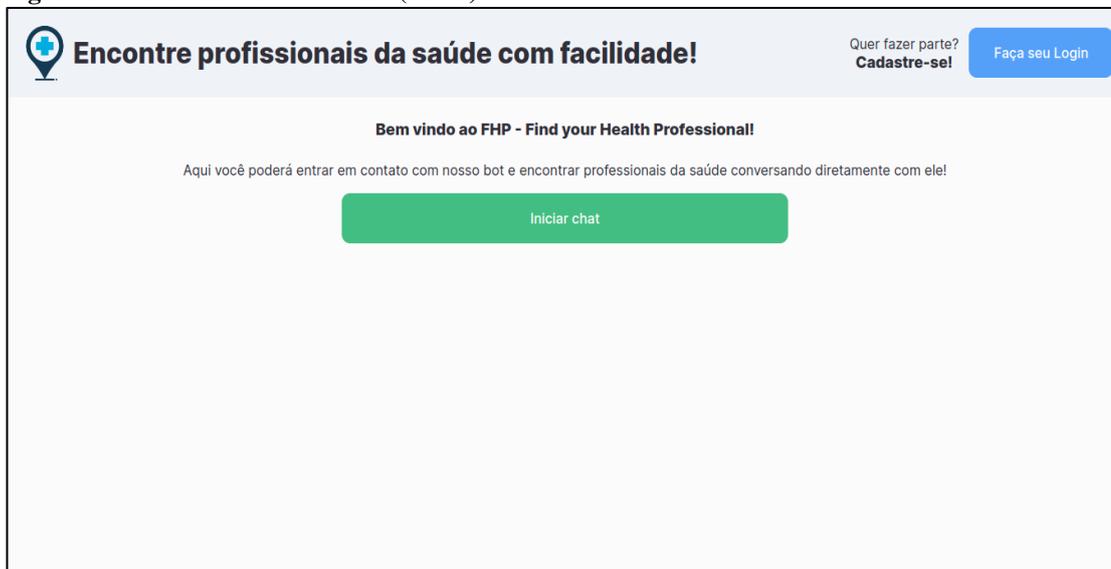
#### 4.6.2 Funcionamento

Nesta seção será feito um detalhamento maior sobre o funcionamento da aplicação, através da descrição das funcionalidades e a utilização de imagens do sistema finalizado, com as funcionalidades já implementadas.

O protótipo trata-se de uma aplicação web responsiva, e visa atender dois tipos de usuários, os que querem fazer parte da plataforma pois são profissionais da saúde e aqueles que desejam buscar um profissional através do chat. Uma pessoa que deseja fazer parte da plataforma tem acesso ao seu painel de dados, onde ele pode administrar seus dados pessoais, suas especialidades e os respectivos locais de atuação. Já o usuário anônimo tem acesso apenas a tela inicial do sistema, aonde ele pode iniciar um bate-papo com o robô e buscar por profissionais da saúde que estão cadastrados na plataforma.

Contudo mesmo que o usuário esteja logado no sistema ele tem acesso ao bate-papo (CSU04), pois a tela inicial é pública, conforme está representada na Figura 15.

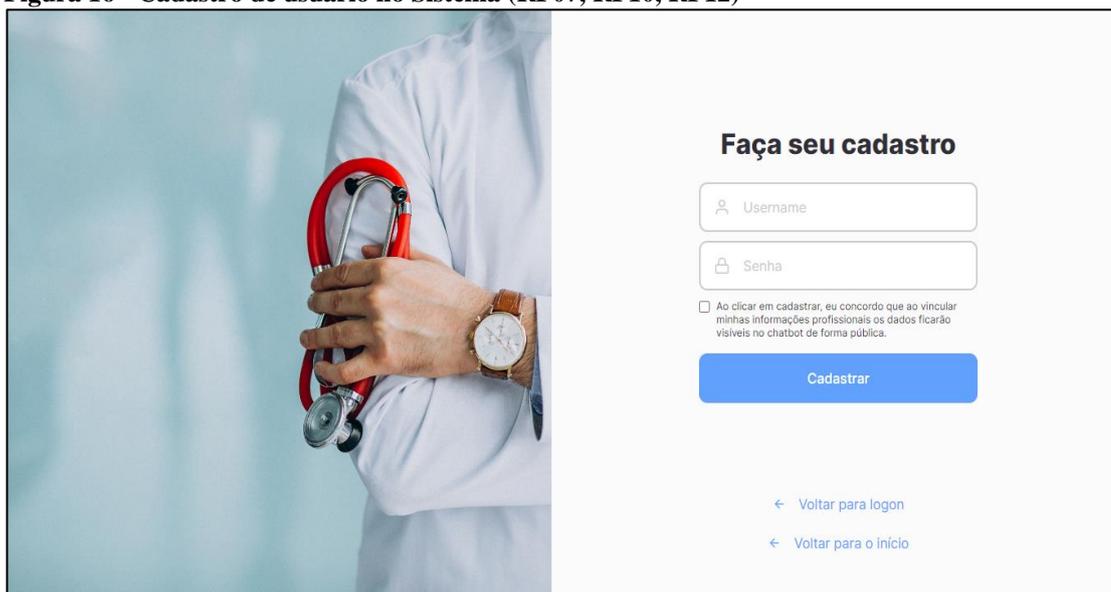
**Figura 15 - Tela inicial do sistema (RF02)**



Fonte: Acervo do Autor.

Conforme citado anteriormente, para o profissional fazer parte do sistema ele deve inicialmente realizar seu cadastro de usuário (CSU01). Para realizar o cadastro, o usuário deverá aceitar os termos de uso da plataforma (CSU17). A Figura 16 representa a tela de cadastro do sistema.

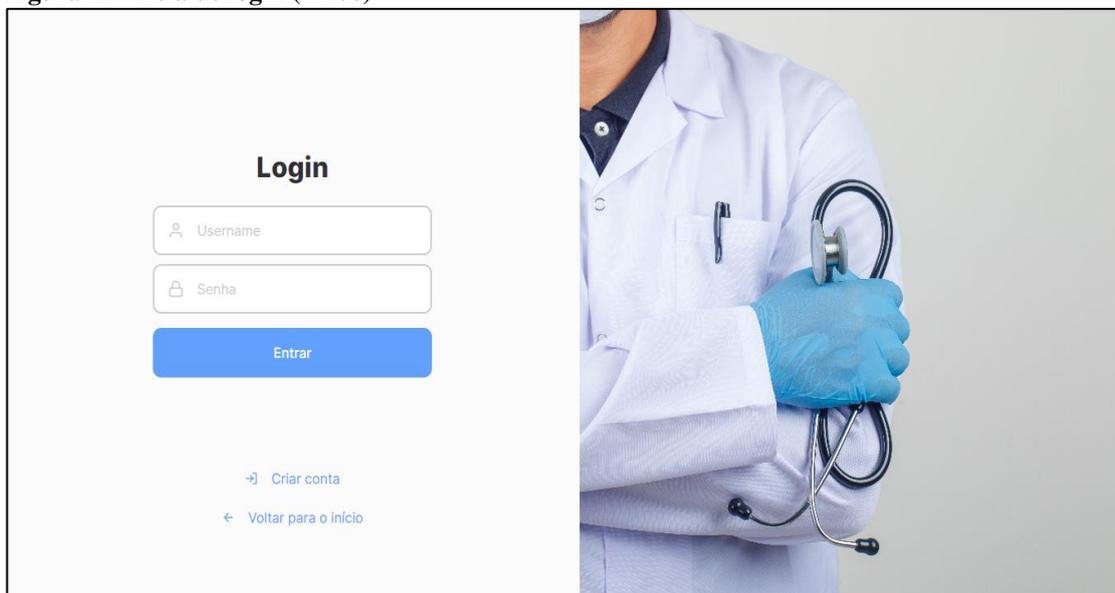
**Figura 16 - Cadastro de usuário no Sistema (RF07, RF10, RF12)**



Fonte: Acervo do Autor.

As informações do cadastro serão validadas no banco de dados, ou seja, validando se o nome de usuário e senha são válidos. Após o cadastro, o usuário é redirecionado para a tela de login onde pode acessar seu perfil. A Figura 17 representa a tela de login do sistema.

**Figura 17 - Tela de login (RF06)**



Fonte: Acervo do Autor.

Para que o usuário possa inserir as suas especialidades e os respectivos locais de atuação, é obrigatório que ele preencha seus dados pessoais primeiro (CSU05). Após o cadastro, o profissional não pode mais alterar seu CPF (RF03), contudo as outras informações ficarão disponíveis para alteração a qualquer momento. Através da Figura 18 será possível visualizar o formulário de cadastro de dados pessoais no primeiro acesso do usuário.

**Figura 18 - Formulário de cadastro de dados pessoais no primeiro acesso do usuário (RF01)**

The screenshot shows a web interface for a doctor's registration. On the left is a sidebar with a 'Voltar para o início' link at the top, followed by menu items: 'Dados Pessoais' (highlighted), 'Especialidades', and 'Locais de Atuação'. At the bottom of the sidebar is a red 'Logout' button. The main content area is titled 'Bem vindo doutor.lucas' with the subtitle 'Continue seu cadastro de dados pessoais'. It contains a form with the following fields: 'Nome completo', 'CPF', 'E-mail', 'Telefone' and 'CEP' (with a right arrow), a state selection dropdown ('Selecione um estado...'), a city selection dropdown ('Selecione uma cidade...'), 'Endereço', 'Bairro', and 'Complemento'. A green 'Salvar' button is located at the bottom right of the form.

Fonte: Acervo do Autor.

Após o usuário finalizar seu cadastro de dados pessoais, o usuário pode vincular suas especialidades profissionais (CSU07), conforme ilustrado na Figura 19.

**Figura 19 - Tela de Especialidades do Profissional (RF11, RF14)**

The screenshot shows a screen titled 'Especialidades'. The sidebar is identical to Figure 18, with 'Especialidades' highlighted. The main content area displays a list of three existing specialties, each in a grey box with a trash icon on the right: 'Dentista' (CRO-123456), 'Dentista' (CRM-000000), and 'Dentista' (CRM-111111). Below the list are four input fields: a dropdown for 'Selecione uma especialidade...', a dropdown for 'Selecione uma entidade de classe...', a text field for 'Número do registro', and a green checkmark icon. A green 'Nova especialidade' button is at the bottom right.

Fonte: Acervo do Autor.

O usuário também pode vincular os seus locais de atuação (CSU09) a várias especialidades (CSU10), tal qual informar quais métodos de pagamento ele aceita (CSU11) e os convênios médicos de cada local. Através da Figura 20 será possível visualizar o formulário de cadastro de um novo local de atuação.

**Figura 20 - Formulário de Cadastro de um Local de Atuação (RF18)**

Fonte: Acervo do Autor.

Além disso, o usuário pode gerenciar seus locais de atuação na tela de consulta de dados (CSU14), onde é possível além de criar um novo registro, editar (CSU15) ou deletar dados já existentes (CSU16), como é possível visualizar na Figura 21.

**Figura 21 - Consulta de Locais de Atuação (RF13)**

Estado	Cidade	Endereço	Ações
Santa Catarina	Rio do Sul	Rua XV de Novembro, 100	<a href="#">Editar</a> <a href="#">Excluir</a>

Fonte: Acervo do Autor.

Já na página principal, que é acessível tanto para usuários anônimos quanto para usuários logados, após iniciar um chat com o robô (CSU04), é possível solicitar informações de profissionais em determinadas regiões e para determinadas especialidades.

A Figura 22 representa uma conversa entre robô x usuário em um fluxo de busca por profissional da saúde (CSU12).

**Figura 22 - Conversa de robô x usuário (RF05, RF09, RF16)**

Olá! Para realizar a busca de profissionais, insira a região desejada nos campos abaixo:

SC - Santa Catarina

Rio do Sul

Continuar

Por favor, me informe qual o segmento da saúde desejado.

Quero um dentista

Certo, aqui estão os dentistas que eu encontrei:

Lucas	Cleber
📍 Santa Catarina - Rio do Sul	📍 Santa Catarina - Ri
<b>Endereço:</b> Rua XV de Novembro, 200, Centro - 88400-000	<b>Endereço:</b> Rua XV Nov
<b>Telefone:</b> 35331500	<b>Telefone:</b> 47 3533-15
<b>Convênios:</b> UNIMED	<b>Convênios:</b> SUS, UNIM
<b>Métodos de Pagamento:</b> Dinheiro, Cartão de Débito, Cartão de Crédito, Bolet	

Enviar

Fonte: Acervo do Autor.

Desta forma, conclui-se o desenvolvimento do protótipo de aplicação. Na próxima sessão serão abordados assuntos referentes as considerações finais do projeto.

## 5. CONSIDERAÇÕES FINAIS

O trabalho constitui-se no desenvolvimento de um protótipo de aplicação para buscar profissionais da saúde através de um *chatbot*, permitindo que os profissionais venham até a plataforma e cadastrem-se para fazerem parte das buscas.

As tecnologias utilizadas para a criação do protótipo, sejam elas ferramentas ou linguagens de programação, ofereceram um grande auxílio em todo o processo de desenvolvimento e permitiram com que o protótipo pudesse ser concluído.

No *back-end* do servidor principal o Node.js e as bibliotecas Express.js e TypeORM ofereceram muita agilidade para o processo de desenvolvimento, permitindo a implementação de algumas metodologias do SOLID, onde foi possível a utilização de *use cases*. Além disso, o banco de dados PostgreSQL permitiu com que os dados se mantivessem íntegros e seguros na aplicação.

Referente aos servidores do chat e do *bot*, o Node.js em conjunto com a biblioteca *Node NLP* permitiram a criação de um treinamento fácil e rápido para o robô, além de remover essa responsabilidade do servidor principal, fazendo com que toda a programação do robô e do chat ficassem isoladas em projetos separados – mas que ainda fazem parte do projeto como um todo – facilitando o desenvolvimento e a manutenção futura.

Quanto ao desenvolvimento do *front-end*, o React.js ofereceu grande agilidade devido a sua *componetização*, fazendo com que cada item da tela tenha a sua responsabilidade e dessa forma, facilita na criação e na manutenção do projeto. Além disso, o uso de componentes externos criados por outras pessoas para o React também foi de grande auxílio.

Em relação aos objetivos específicos do trabalho, em relação ao que propunha a levantar requisitos para desenvolver o *chatbot*, este foi cumprido e está detalhado na seção de requisitos e na seção de regras de negócio. Além disso, fora realizado todo projeto de engenharia de software que visava implementar o protótipo seguindo as especificações dos requisitos.

Ainda sobre o desenvolvimento do protótipo, o objetivo que visava definir o método de configuração da linguagem natural do robô obteve sucesso, e foi descrita na seção de revisão da literatura no tópico sobre *Node NLP*.

No tópico de Diagramas foi descrito diversos diagramas sobre o protótipo, dentre eles os casos de uso, fluxogramas, distribuição de aplicação e entidade-relacionamento. O último citado cumpre o objetivo que buscava modelar a estrutura de dados para armazenar no PostgreSQL.

Além desses citados, antes do início do seu desenvolvimento, foi realizada uma pesquisa para identificar ferramentas já existentes no mercado, que possuem uma finalidade semelhante e explicar seu funcionamento, onde através da seção de Estado da Arte, foi apresentado os aplicativos Uibem para pacientes e Uibem para profissionais. Não só isso, mas também foram criados todos os protótipos das telas do sistema antes através do Figma, e com isso foi possível definir toda a interface de operação do usuário.

O protótipo desenvolvido oferece a qualquer pessoa que entre em contato com o robô e busque por profissionais cadastrados na plataforma na região que desejar, e quando encontrado pode pegar os dados de cada profissional para realizar algum objetivo desejado.

Já o profissional que deseja participar da plataforma pode cadastrar suas inúmeras especialidades e detalhar todos seus locais de atuação, junto com meios de pagamentos, contato e convênios médicos, para dessa forma o robô mostrar seu cartão de visita para os usuários que realizarem buscas.

Desta forma pode se concluir que os objetivos propostos para o trabalho foram alcançados com sucesso. Entretanto, considerando as limitações deste trabalho, a seguir serão listadas algumas recomendações para trabalhos futuros, incluindo melhorias e novas funcionalidades para o protótipo.

## 5.1 RECOMENDAÇÕES DE TRABALHOS FUTUROS

Como recomendação de continuidade desse projeto, uma delas seria a implantação de novas funcionalidades para o profissional da saúde para que haja um painel de solicitações de agendamentos dentro de seu perfil, enquanto no chat, o usuário que busca um profissional pode através do cartão de visita do profissional realizar uma solicitação de consulta.

Além disso, adicionar mais regras de negócio envolvendo o profissional para evitar que pessoas que não são profissionais de verdade entrem na plataforma, seja mais validações em volta do número do registro ou até mesmo uma solicitação de aprovação, aonde o administrador do sistema busca validar se o profissional é válido ou não para participar da plataforma.

Para que o sistema fique mais completo e possa se receber diversos profissionais de diversas áreas diferentes da saúde, a adição de mais áreas e ramificações da área da saúde seria importante, fazendo com que o sistema aceite desde médicos, enfermeiros ou veículos particulares (como ambulâncias), e com isso o robô pode se adaptar a diversos contextos desejados pelo usuário.

Realizar melhorias no conhecimento do robô também será muito importante para que ele esteja apto a atender todas as melhorias citadas anteriormente, com isso seria necessário adicionar mais intenções e palavras chaves para aumentar o poder de processamento de linguagem natural do robô, e se for necessário, realizar a troca da biblioteca buscando um utilitário mais completo do que o Node NLP.

Por fim, mas não menos importante, poderá ser realizado alguns testes com um grupo de usuários, a fim de validar o sistema e sua funcionalidade.

## REFERÊNCIAS

- ALMEIDA, Flávio. **Mean: Full stack Javascript para aplicações web com MongoDB, Express, Angular e Node**. Editora Casa do Código, 2015. Ebook.
- ALVES, William Pereira. **HTML & CSS: aprenda como construir páginas web**. São Paulo: Expressa, 2021. Ebook.
- ANDRADE, Ana Paula de. **O que é Express.js?**. 2021. Disponível em: <https://www.treinaweb.com.br/blog/o-que-e-o-express-js>. Acesso em: 31 maio 2022.
- AXA GROUP. **Nlp.js**. 2022. Disponível em: <https://github.com/axa-group/nlp.js>. Acesso em: 15 nov. 2022.
- BRASIL. **Lei Geral de Proteção de Dados Pessoais (LGPD)**. 2021a. Disponível em: <https://www.gov.br/cidadania/pt-br/acao-a-informacao/lgpd>. Acesso em: 17 ago. 2022.
- BRASIL. **Princípios da LGPD**. 2021b. Disponível em: <https://www.gov.br/cidadania/pt-br/acao-a-informacao/lgpd/principios-da-lgpd>. Acesso em: 17 ago. 2022.
- COMER, Douglas E. **Interligação de Redes com TCP/IP: Princípios, Protocolos e Arquitetura**. 6. ed. Elsevier, 2015. v. 1. Ebook.
- DUARTE, Luiz. **O que é Node.js e outras 5 dúvidas fundamentais**. 2021. Disponível em: <https://www.luiztools.com.br/post/o-que-e-nodejs-e-outras-5-duvidas-fundamentais>. Acesso em 31 maio 2022.
- FLANAGAN, David. **Javascript: o guia definitivo**. 6. Porto Alegre: Bookman, 2014. Ebook.
- FOROUZAN, Behrouz A. **Protocolo TCP/IP**. 3. Porto Alegre: ArtMed, 2010. Ebook.
- FREITAS, Pedro Henrique et al. **Programação Back End III**. Porto Alegre: Sagah, 2021. Ebook.
- DA REDAÇÃO. **Brasileiros vandalizam inteligência artificial e app é retirado do país**. 2018. Disponível em: <https://www.gazetadopovo.com.br/economia/nova-economia/brasileiros-vandalizam-inteligencia-artificial-e-app-e-retirado-do-pais-29x2pje9tcxbkijhaainuhb22/>. Acesso em: 3 dez. 2022.
- IUBEM. **Iubem – Seu app de saúde**. Disponível em: <https://iubem.com>. Acesso em: 06 set. 2022.
- KOVACS, Leandro. **O que é processamento de linguagem natural? [NLP]**. 2022. Disponível em: <https://tecnoblog.net/responde/o-que-e-processamento-de-linguagem-natural-nlp/>. Acesso em: 01 junho 2022.
- LOPES, Jean de Oliveira. **PHP ou Typescript: uma comparação de duas linguagens para web pelas suas características**. Porto Alegre, 2017.

MARCHI, Kevin. **Catálogo Médico: Desenvolvimento de Protótipo de Aplicação Web Responsiva**. 2021. 70 p. Trabalho de Conclusão de Curso (Bacharelado em Sistemas da Informação) - Centro Universitário para o Desenvolvimento do Alto Vale do Itajaí, Rio do Sul, 2021. Disponível em:

<https://www.unidavi.edu.br/bibliotecatrabalhos/consultartrabalho/trabalho/65490/arquivo/65489/download>. Acesso em: 6 dez. 2022.

MDN CONTRIBUTORS. **Começando com React**. 2021a. Disponível em: [https://developer.mozilla.org/pt-BR/docs/Learn/Tools\\_and\\_testing/Client-side\\_JavaScript\\_frameworks/React\\_getting\\_started](https://developer.mozilla.org/pt-BR/docs/Learn/Tools_and_testing/Client-side_JavaScript_frameworks/React_getting_started). Acesso em: 6 maio 2022.

MDN CONTRIBUTORS. **Métodos de requisição HTTP**. 2021b. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Methods>. Acesso em: 30 maio 2022.

MELO, Diego. **O que é Node.js? [Guia para iniciantes]**. 2021. Disponível em: <https://tecnoblog.net/responde/o-que-e-node-js-guia-para-iniciantes/>. Acesso em 31 maio 2022.

MILETTO, Evandro M.; BERTAGNOLLI, Silvia de C. **Desenvolvimento de software II: introdução ao desenvolvimento web com HTML, CSS, Javascript e PHP**. Porto Alegre: Bookman, 2014. Ebook.

OKUBO, Beatriz. **Você sabe o que é CSS? Entenda como funciona e para que serve**. 2021. Disponível em: <https://br.godaddy.com/blog/voce-sabe-o-que-e-css-entenda-como-funciona-e-para-que-serve/>. Acesso em: 01 junho 2022.

OLIVEIRA, Cláudio Luís V.; ZANETTI, Humberto Augusto P. **Node.js: programe de forma rápida e prática**. São Paulo: Expressa ou Editora Saraiva, 2021. Ebook

OLIVEIRA, Cláudio Luís V.; ZANETTI, Humberto Augusto P. **Javascript descomplicado - programação para web, IOT e dispositivos móveis**. São Paulo: Erica, 2020. Ebook.

ORACLE. **O Que É um Banco de Dados?**. 2022. Disponível em: <https://www.oracle.com/br/database/what-is-database/>. Acesso em: 31 maio 2022.

PRESSMAN, Roger, S.; MAXIM, Bruce R. **Engenharia de software: uma abordagem profissional. 9. ed.** Porto Alegre: AMGH, 2021. Ebook.

RED HAT. **O que é API REST?**. 2020. Disponível em: <https://www.redhat.com/pt-br/topics/api/what-is-a-rest-api>. Acesso em: 31 maio 2022.

ROVEDA, Ugo. **React: O que é, como funciona e porque usar e como aprender**. 2020. Disponível em: <https://kenzie.com.br/blog/react/>. Acesso em: 31 maio 2022.

SAUDATE, Alexandre. **REST: Construa APIs inteligentes de maneira simples**. Editora Casa do Código, 2014. Ebook.

SILVA, Douglas da. **Processamento de linguagem natural: entenda como funciona, importância e aplicação [Guia Completo]**. 2021. Disponível em: <https://www.zendesk.com.br/blog/processamento-de-linguagem-natural/>. Acesso em: 01 junho 2022.

SOARES, Wallace. **PHP 5: Conceitos, programação e integração com banco de dados. 7.** São Paulo: Erica, 2013. Ebook.

SOMMERVILLE, Ian. **Engenharia de Software. 9. ed.** São Paulo: Pearson Education do Brasil, 2011. Ebook.

SOUZA, Ivan de. **PostgreSQL: saiba o que é, para que serve e como instalar.** 2020. Disponível em: <https://rockcontent.com/br/blog/postgresql/>. Acesso em: 31 maio 2022.

SOUTO, Mario. **O que é front-end e back-end?**. 2019. Disponível em: <https://www.alura.com.br/artigos/o-que-e-front-end-e-back-end>. Acesso em 31 maio 2022.

SERPRO, **O que são dados anonimizados, segundo a LGPD.** 2022. Disponível em: <https://www.serpro.gov.br/lgpd/menu/protecao-de-dados/dados-anonimizados-lgpd>. Acesso em: 29 set. 2022.